Issue 19 | Nov 2008

Blender learning made easy

blender art MAGAZINE

Rigging & Constraints

Tutorial - To Rig a Boat's Rigging

Tutorial -Perpetual Motion Machine Rigging

Tutorial - Caterpillar Soft Tracks

Tutorial - Animation Temporal Verification

COVERART -Moonflower - by Derek Watts

blender art MAGAZINE

EDITOR
Gaurav Nawani gaurav@blenderart.org

MANAGING EDITOR
Sandra Gilbert sandra@blenderart.org

WEBSITE
Nam Pham nam@blenderart.org

DESIGNER
Gaurav, Sandra, Alex

PROOFER
Kevin Braun
Phillip Ryals
Bruce Westfall
Joshua Leung
Lynda Schemansky
Eric Pranausk
Noah Summers
Joshua Scotton
Mark Warren
Wade Bick
Patrick O'Donnell
Brian C. Treacy
Scott Hill
Henriel Veldtmann

WRITERS
Johannes Ewers
William Edstrom
Hamed Zaghaghi
Alain Mathez

COVER ART
Moonflower - by Derek Watts
dwatts1@gmail.com

CONTENTS

Sandra Gilbert
Managing Editor

"While I am still not a rigging master, I no longer stare in confusion at the amazing mess I just made of my rig. And better yet, the rigs actually work... well kind of. :P "

Very few of us enjoy rigging a character, but that is no excuse for not learning how to set up at least a basic rig. Especially considering the number of great resources that have become available over the last couple of years.

In addition to the large number of tutorials written by various community members and all the free rigs available for study, there are two great books (Introducing Character Animation with Blender and Animating with Blender) as well as Bassam's ManCandy FAQ's DVD available for reference and study.

After reading (and re-reading) both books and watching Bassam's ManCandy DVD (too many times to count), the confusion began to clear as I started seeing how it all went together. So I finally gave up avoiding character rigging. While I am still not a rigging master, I no longer stare in confusion at the amazing mess I just made of my rig. And better yet, the rigs actually work... well kind of. :P I think I still need to keep practicing.

The moral of this long winded introduction: if I can learn to rig, so can you. Which brings us to the theme for this issue of Blenderart Magazine. We are taking a look at various rigging techniques. Some of which actually have nothing to do with characters.

Surprise! Armatures and constraints aren't just for rigging characters. They can be used to create an animation rig for any object that moves, or that you would like to have move. And we have some pretty inventive rigs/set ups to share.

So go grab a hot drink (I don't know about you but it is cold where I am) and settle down for a very informative read.

sandra@blenderart.org

Well okay, yes it is still complicated, but there is a logic, flow and even a pattern to it.

Introduction

For years I have avoided anything beyond very simple rigging. In fact in my effort to avoid learning what seemed like an impossibly complicated skill, I ended up becoming the queen of the incredibly simple rig. My rigs had no constraints, no bone shapes and as few bones as possible. Then again, I was only posing characters for still images, there was no animating controls needed.

Trying to study and learn from the variety of free rigs available only added to my confusion. They were composed of so many bones, constraints and odd bone shapes, I never could make heads or tails of just what was going or how to build one of my own.

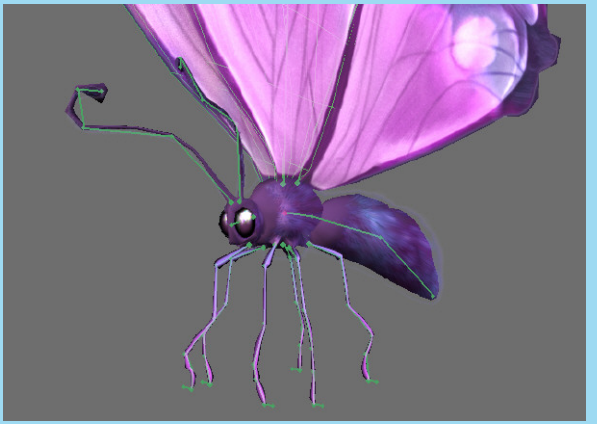
But after serious study of some great resources, I realized that the seeming chaos of rigs is actually not that complicated. Well okay, yes it is still complicated, but there is a logic, flow and even a pattern to it.

When you break it down to the most basic idea, a good working rig has three main components.

- Controls
- Deformers
- Helpers

That seems simple enough, and yet this is exactly where it starts to seem complicated. So let's look at what each component does and see if we can't clear this up a little bit more.

Controls are the parts of the rig that you actually select to do any animating. It helps to remember that not every bone in a rig is actually directly manipulated during animation. Controls can be very simple (plain armature bones) to very fancy and complex bone shapes that visually tell you what it does.



Make sure all your controllers are part of the same armature object.

Helpers are not manipulated directly and generally hidden during the animation process.

Helpers are kind of like the frame you build when you build a house, you definitely need them, but you don't do much with them once they are put in place.

Some examples of **Helpers**:

- leg bones
- Spine
- Arm bones

Deformers are driven by the Controllers to do the actual work. Like the Helpers, at animation time, they are not seen by the animator.

Some examples of Deformers:

The armature itself is considered to be a deformer

Mesh Deform Modifier

- Shape Keys
- Lattices
- Hooks

So now that we know what makes up a rig, where and how do you start? We won't be building a full rig here (there are plenty of excellent tutorials already written about that). Instead we look at a good workflow. Here are some tips to get you started:

Probably the best advice for beginning riggers is to build your rig in stages. That way you won't forget something and you can fix problems as you encounter them.

- It is best to start by creating a simple armature setup.
- Following the general shape of your character, add bones to create a simple stick figure
- Make sure all your Parent relationships are set up and correct.
    - i.e. Hand bones are parented to a lower arm bone which is parented to an upper arm bone etc.
    - Make sure you don't have any bones that get left behind when you move the armature.
    - Test all movements.

Remember: to enter object mode and use Ctrl + A to apply scale / rotation before setting up any IK's, Contraints or Controllers.

- Take it one section at a time.
    - You can start from the top of your rig and work down, from the bottom up, one limb at a time, or from the spine out, just don't bounce around willy nilly. You will end up forgetting something.
- Start setting up IK solvers and any needed constraints.
    - (i.e. pole targets, track to, etc.)
    - If setting up right leg, next go to left leg and make sure settings all match
- Set up controllers: These can include controls to operate the shape keys for facial animation, controls for curling fingers, standing on tip toe (foot) as well as many others.
    - You can set the controllers as you are setting up the IK solvers and Constraints or in a separate 2nd sweep through the rig
- Test all bones, joints and controls.
- Bind the rig to the character using the Armature Modifier (or to a Mesh Deform Modifier that will be deforming your character).
    - Test all bones, joints, controls for pinching, distortion etc.
- Time to separate the rig into bone layers for a cleaner look for the animator.
    - All controls on one layer
    - All helpers on one or more layers (you can separate them into various layer groupings for

  - for organizational reasons if wanted or put all helpers on one layer

- Add a Master (often called Root) bone: everything in the rig will be connected in one way or another to this bone.
  - Any bone that is not a child of another bone will be directly parented to the Master bone
  - Test all bones to see that everything works well

While this was not meant to magically turn you into a master rigger, hopefully it took some of the mystery and confusion out of creating a good rig.

Great Rigging Resources

Introducing Character Animation with Blender

Animating with Blender

ManCandy FAQ's DVD

BSoD: Introduction to Rigging

Blender Gamekit 2nd edition

Carsten Wartmann, writer of many Blender books and main author of the Blender Gamekit, has started work on a fully updated version of the Blender Gamekit! This project, including DTP/design by Samo Korosec, is scheduled to finish during November. So it should be in time for Santa to deliver it under the Christmas trees.

The updated Gamekit will cover all new functionality, GLSL materials, multilayer texture, character animation and logic states and bricks. Carsten will also clean-up the old tutorials to make them work well, and add new ones, also based on Yo Frankie!

The Blender Gamekit book update: Project is now in its last stage. We're reviewing the designed pages, finishing the CD for the book, and hopefully within one or two weeks it goes to print! (But you know, planning doesn't always match human limitations!)

You can now pre-order copies even up to December 15 with a nice 20% discount!

Needless to say, revenues will help Blender Foundation projects (and me & Brecht coding 2.5)!

-Ton-

Yo Frankie! DVDs

The Yo Frankie DVDs are finished and shipping has started. If you haven't ordered a copy yet, you can still order one from the Blender e-shop. It would make a great gift for the holidays.

Great Holiday Gift Ideas

The holidays are fast approaching, but there is still time to pick up a little something for yourself or that special Blenderhead on your gift list.

Creature Factory

The ManCandy FAQ

Big Buck Bunny DVD

Bundle: Essential + BBB

Bounce, Tumble and Splash!

BlenderNewbies Video Tutorial Compilation DVD-ROM

Game Competition

This competition is being run to encourage developers (or teams of developers) to create fun, cool looking content for the Game Engine built into Blender.

With the release of 2.48, the GE has received a HUGE boost in terms of speed, functionality, and more importantly, kudos.

Major Categories

Euro400 of the prize fund will be split into two categories...

Category 1: Best Graphics

Prize: Euro200

This prize will be awarded to the best use of graphics. While using GLSL might seem to be the best way to achieve this for some effects, if someone enters a great looking entry that doesn't use GLSL, this will be taken into account during the voting.

Entries could be walk throughs, graphical demos as in scene.org, and of course full on games.

Category 2: Best Game play

Prize: Euro200

This prize will be awarded to the game that is the most fun to play.

Anything goes (well, check the few rules below). You can have more than one team member, you can use GLSL, you can use external libraries such as PyGame. As long as the 3D engine being used is the Blender GE engine, and the game will run from within Blender by pressing P, then your entry will be valid.

Minor Categories

Note: These categories do not exclude entry to the above main categories.

Category 1: Best content created using only logic bricks (no Python scripting).

Prize: Euro100

This prize will be awarded to the best content created using just Logic Blocks. The Blender GE Logic Block system is a very powerful way for artists with no coding skills to create interactive content. This prize awards artists using Blender in this way.

For further information and rules regarding this contest, visit the Game Competition Announcement page.

Introduction

A sailing boat or ship is a complicated vehicle consisting of hundreds of moving objects like sails, ropes, blocks, and booms. Depending on the direction of the wind and the direction of the boat, all moving parts take different positions in relation to each other. If you plan to create a boat model that is not static, you should include some automation capabilities into your rig. Otherwise, you will have to change many objects every time you re-position the sails. Blender's constraints and armatures provide excellent tools to improve efficiency when modifying a boat's rigging.

This tutorial will explore some basic concepts of automating a boat's rigging with the help of a simplified rig. Out of scope is a realistic detailed boat model. That is up to you.

Planning the boat model

For our tutorial, we use a small boat consisting of a hull, a single mast, and a sail that is controlled by a boom on the lower seam and a gaff boom on the upper seam. The boom and gaff boom are connected to the mast by hinges allowing a rotation around the z-axis. The boom is fixed into its position by a tackle. Sail and gaff boom are hoisted into their position by another tackle.

Where do we need automation in a boat's rigging? Have a look at our little boat model. If the wind comes from a forward direction, the angle between boom and boat length axis is small and the sail is rather flat. If the wind comes from the side, we have to rotate the boom into a position with a larger angle (45 degree).

The sail transforms into a more convex form. The gaff boom and its tackle move into a new position. If you look at the tackle, you see that upper and lower block have changed rotation and position. The ropes between the blocks stretch over a longer distance. To make all these modifications manually is a tedious and time-consuming task.

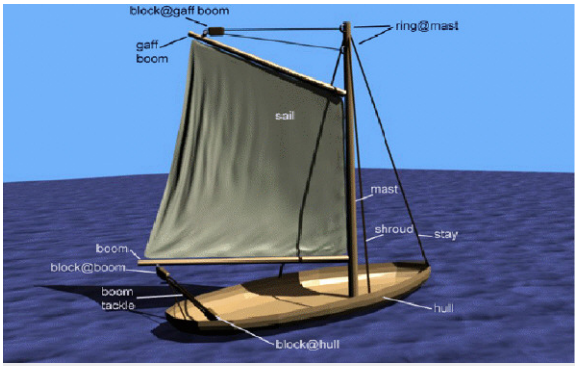


Figure 1: basic boat model

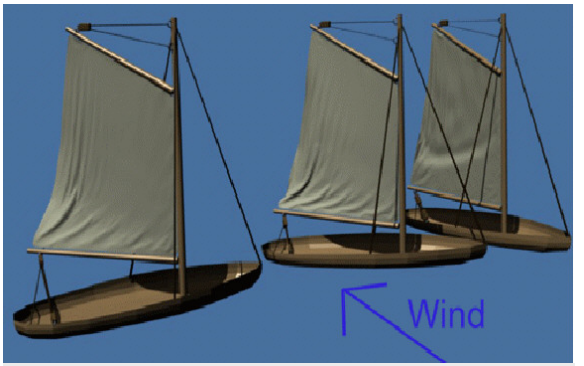


Figure 2: rigging adapts to wind

By Johannes Ewers

Our goal is to automate the movement of tackles and sail so that they follow the rotation of the two booms. The form of the sail should adjust in a natural way.

To do that we need the following automated behavior:

- The block at the gaff boom, (the upper boom) follows the boom and rotates in a way so that its free end always faces the mast.
- The ropes between mast (fixed end) and gaff boom block follows the movement.
- The block at the (lower) boom faces its counterpart that is fixed to the hull.
- The block that is fixed to the hull always faces the block at the boom.
- The ropes between the two blocks follow the movement and stretch or shrink depending on the blocks' positions.
- The lower seam of the sail follows the boom; the upper seam follows the gaff boom.
- The sail is always stretched between both booms.
- The sail shows wrinkles depending on the direction and strength of the wind.

Building the standing rigging

We first need a small boat for our rigging experiments. I took a plane object, extruded the front edge three times, and formed the resulting mesh into a rough boat shape. Then I extruded the outer edges two times on z-axis. This provides us with a simple hull.

On the hull, I then set a thin cylinder object as mast. Another very thin cylinder object is positioned between bow and mast top representing a stay used on a ship to stabilize the mast. Another two cylinders are positioned left (port side) and right (starboard) from the mast as shrouds for additional stabilization. Mast, shrouds, and stay form the "standing rigging" of our boat. That is the collection of the fixed, non-moveable parts. Now we need three ring objects as connection between fixed and movable parts of the rigging.

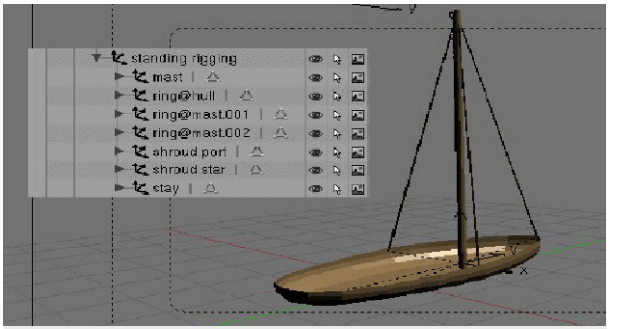


Figure 3: standing rigging of boat

Exercise; create a ring:

- Start with creating a circle. Add››Mesh››Circle, Vertices:8 should be enough.
- Switch to side view. You see the circle as a thin line from the side.
- Select the circle object.
- Switch to Edit Mode. Select all vertices of the circle.
- Move the 3D-cursor to a suitable position for the middle of the ring.

by Johannes Ewers

by Johannes Ewers

- Open the Mesh Tools panel in a Buttons Window.
- Enter Degr: 360, Steps: 8, Turns: 1 and press Spin.
- The cursor changes to a "?" symbol. Click on the Side View window.
- Scale the resulting ring object to suitable size, about half of the diameter of the mast.

We need two rings at the top of the mast (see figure 6) and one ring at the bottom of the hull near the stern (rear-end of the boat).

Now it's time to put some structure and hierarchy into the model. We create an empty *(Add››Empty)* and name it "standing rigging". We move the empty to a location near to or on the mast. We create a compound object by "parenting" all parts of the standing rigging to the empty. Therefore, select mast, stay, shrouds, and rings and as last object, the empty. The Outliner Window is very useful for this task. Now select the function *Object››Parent››Make* Parent to create a sub-structure in the Outliner. Create another Empty, name it "little boat", and move it to the middle of the hull mesh. Create a compound object by "parenting" the hull mesh and the standing rigging to the "little boat" Empty.

Now you can position and rotate the boat just by selecting and moving the parent "little boat". That was the easy part.

Building the running rigging

The "running rigging" is the collection of all moveable booms, blocks, and ropes on the boat.

a) Create the block objects

We start with building three blocks that we need for the running rigging. A block is part of a tackle. It consists of a ring and a cube we call block cage that would house the wheels of a tackle.

- Create a ring as described above.
- Create a cube (Add››Mesh›Cube) and name it "block cage"
- Switch to Edit Mode and select all vertices of the cube.
- Use Mesh››Edges››Bevel to round the edges of the cube. That looks nicer.
- Now scale the cube as shown in figure 4.
- Move the ring to the bottom of the cube.
- Create an empty, name it "block" and move it to the center of the ring. The position/center of the empty is very important. It determines the rotation pivot of the block!
- Create a compound object by parenting the cube and the ring to the empty.

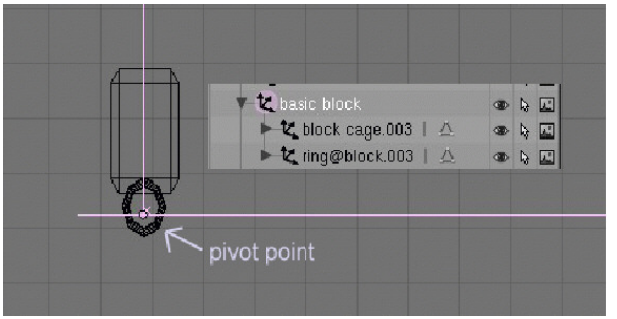


Figure 4: basic block assembly

- Now we need three of these blocks, two with the ring on the bottom side and one with the ring on the top side.
- Select the block compound object in the Outliner. Use *Select››Grouped››Children* to select all objects in the compound. This is import for copying the whole compound and not just the parent object.
- Use *Object››Duplicate* to create a second block Repeat this for the third block. Name the objects as shown in figure 4.
- Now, in the last block select the "block cage" child object and move it downwards until the ring is on top of it. Do not move the ring! We have to make sure that the center of the compound object is still in middle of the ring.

b) Create the booms

Boom and gaff boom are compound objects consisting of a lengthy cylinder and a ring to connect a block.

- For the boom create a cylinder, rotate it into a horizontal position, and scale it as shown in figure 1. One end of the cylinder should touch the mast at a point a little bit over the rim of the hull. Resting position of the boom is parallel to the boats main axis.
- Create a ring and move it to the free end of the boom on the lower side. Now we merge all boom parts into a compound object. Therefore, we create an Empty. We move the empty to a position between the boom end and the mast. The center of the empty will be the pivot point for rotating the boom.
- Do *Object››Parent››Make Parent* to bind the cylinder and the ring to the boom compound. Repeat the above steps for the gaff boom. Make it a little shorter and give it an angle upwards. The ring has to be located to the top side of the gaff boom.

c) Create an armature for the booms

We will move the booms with an armature consisting of three bones. You might ask "what do we need an armature for? We can rotate the boom directly". However, the armature will also be used to move the upper and lower seams of the sail, synchronized to the boom movement.

- Create an armature (Add›Armature), name it "sail guide", move and scale it so that it is in the same position as the boom. The pivot point of the bone should fit to the pivot point of the boom
- Switch to Edit Mode.
- Extrude the bone two times to get three bones in total.
- Click RMB on the middle part (not the tips) of the third, outermost bone.
- Use Armature››Parent››Clear Parent››Disconnect Bone to separate one bone from the others.
- In the Buttons Window, Armature Bones panel click on Hinge so that the bone does not inherit the rotation of its parent.
- Move and scale the bone so that it is in the same position as the gaff boom also matching its pivot points.
- Separate the second bone from the first one as described above and move it into a position between the upper (Gaff boom) and the lower bone (boom). The middle bone will later be used to control the belly of the sail.

by Johannes Ewers

by Johannes Ewers

Now we need to connect the bones to the booms. We cannot use the armature modifier because that only works with meshes. We do not have a mesh, we have an empty as parent in the boom compound! As an alternative, we will use a "copy rotation constraint".

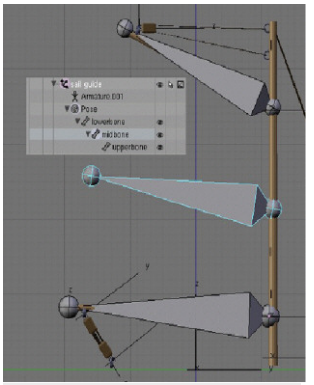


Figure 5: sail guide armature

- In Object Mode select "boom comp" in the Outliner.
- In the Buttons Window, Constraints panel, add a constraint of type Copy Rotation.
- As target, type in the name of the armature "sail guide". As bone, type in "lower bone".
- Choose Z as rotation axis.
- Repeat the same for the gaff boom but choose "upper bone" as target bone.
- To test the setup, select the armature, select "lower bone", switch to Pose Mode, and rotate the bone around its center. The boom should follow the rotation. The same should work for the gaff boom when using the "upper bone".

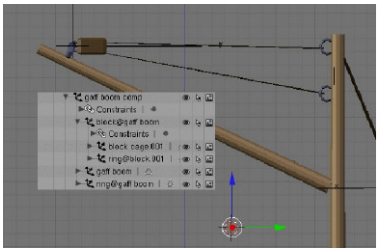


Figure 6: gaff boom assembly

d) Build and automate the gaff boom tackle

In the next step, we will create the tackle that lifts the gaff boom. The tackle consists of a block connected to the gaff boom and two ropes fixed to the mast. Block and ropes should automatically follow the movement of the boom.

- Select a spare block with a ring on the bottom (created in step a).
- Move the block to the ring at the tip of the gaff boom.
- "Parent" the block to the gaff boom.
- On the block use Object››Constraints››Add Constraint››Track To.
- Constraint parameters are: Target:ring@mast.002, To:Z, Up:Y
- Now, the block should orient towards and follow one of the rings at the mast. Test it by rotating the gaff boom (see step c).
- Create a cylinder to model a rope for the tackle and call it "rope@gaff boom.001".
- Switch to Edit Mode. Scale the edges in x/y direction so that a thin tube is the result.
- In Object Mode, the scale factor should still be one for all coordinates. You can check that with the Transform Properties window (Object›› Transform Properties). Sounds strange? The "stretch to constraint" we will use later shows an even stranger behavior if Scale X/Y/Z differ from one.

- Switch to Object Mode. Move the 3D-cursor to the lower end of the cylinder. Use Object››Transform››Center Cursor. That will move the center of the object – the pivot point for rotations – to the lower end.
- Now move the cylinder to one of the rings at the mast without rotating it.
- "Parent" it to the "little boat" compound object.
- Go to the Buttons Window, Object Panels, Constraint and click on "Add Constraint", select "Stretch To". This constraint is only available through the panels.
- Enter Target: block cage.001, Vol:NONE, Plan:Z.
- You might expect the cylinder to orient towards the block but unfortunately, it stays in its position. You first have to correct the position manually.
- Switch to Edit Mode and transform the vertices in the desired way. The rope (cylinder) should stretch between the middle of the block and the ring at them mast. Do not try to do that in Object Mode, it will not work. I took me some experiments to work out the right sequence of steps.
- Now repeat these steps with a second rope (cylinder) between the middle of the block and the other ring.

Now test the gaff boom rig by moving it with help of the upper bone of the "sail guide" armature.

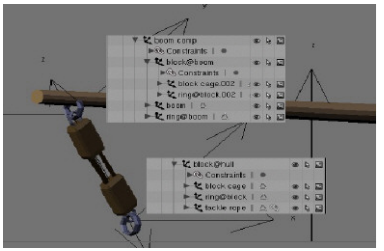


Figure 6: gaff boom assembly

e) Build and automate the boom tackle

We will use the same technique to build the tackle at the boom.

- Take the spare block with the ring on the top and move it to the ring at tip of the boom (see figure 7). "Parent" it to the "boom comp" and name it "block@boom".
- Add a "TrackTo" constraint to the block with Target:ring@hull (part of the standing rigging). The block should now point to the ring.
- Take the spare block with the ring on the bottom and move it to the ring@hull. "Parent" it to the "little boat" and name it "block@hull".
- Between the blocks of a tackle, we have an even number of ropes plus one free rope to pull. Therefore, we create a cylinder, switch to Edit Mode, and scale the cylinder in x/y direction to form a thin tube. Be careful to do all mesh scaling/rotation in Edit Mode and not in Object Mode. The scale factor in Object Mode should be one, otherwise the "StretchTo" constraints will have strange scaling effects. You can check that in the Transform Properties window.
- In Edit Mode duplicate the cylinder mesh three times and shift the mesh elements a little bit to form a kind of cage.
- Switch to Object Mode. Move the 3D-cursor to the lower end of the mesh. Use Object››Transform››Center Cursor. That will move the center of the object – the pivot point for rotations – to the lower end.
- Move the tackle rope mesh to the block at the hull so that the mesh's pivot point matches with the middle of the block.

by Johannes Ewers

- Add a "TrackTo" constraint to the block with Target: block cage.002 (part of the block at the boom). The block and the rope mesh should now point to the other block. Each block will synchronize to the other.
- Now add a "StretchTo" constraint to the rope mesh. Enter Target: block cage.002, Vol:NONE, Plan:Z.
- As a result, the mesh might first point into an unsuspected direction. You have to correct the mesh in Edit Mode and transform it into the correct position and length.
- Now test the boom rig by moving it with help of the lower bone of the "sail guide" armature. The tackle should neatly follow the boom tip and stretch as needed.

Building the sail

We will use two techniques for the sail; mesh modifiers (Armature, MeshDeform) for a basic positioning, a cloth physics simulation for wrinkles and wind deformation.

- Make sure that the booms are in their neutral position parallel to the boat's main axis. To do this, select "sail guide". Switch to Pose Mode and use Pose››Clear Transform››Clear User Transform.
- Start by creating a simple patch (Add››Mesh››Plane). Name it "sail".
- Translate the "sail" plane to the approximate position. Switch to Edit Mode and fit the four corners of the mesh face between boom, gaff boom, and mast. The vertices should all have the same X-coordinate value for a maximal flat mesh. That is helpful for the MeshDeform modifier we will use later.
- Now subdivide the "sail" mesh in Edit Mode four times to create 16x16 faces (Select all vertices, Mesh››Edges›Subdivide).
- We have to define a vertex group later be used to fixate the mesh during the cloth simulation. Therefore, deselect all vertices. Click Shift+RMB on ever third vertex at the seam of the mesh near the booms and the mast starting with a corner.
- Go to Buttons window, Mesh Buttons, Link and Materials, press New for a new vertex group, name it "seam" and press Assign. That will bind the selected vertices of the sail mesh to the vertex.
- Leave Edit Mode.

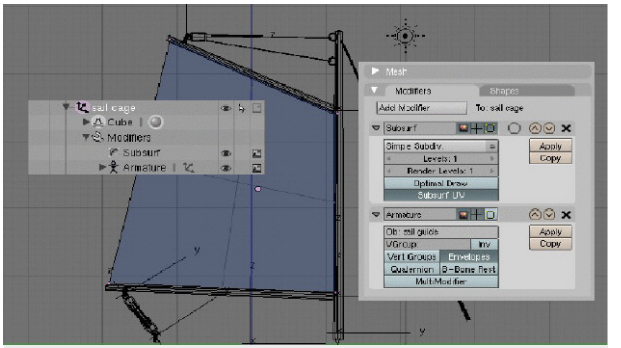


Figure 7: boom and tackle assembly

by Johannes Ewers

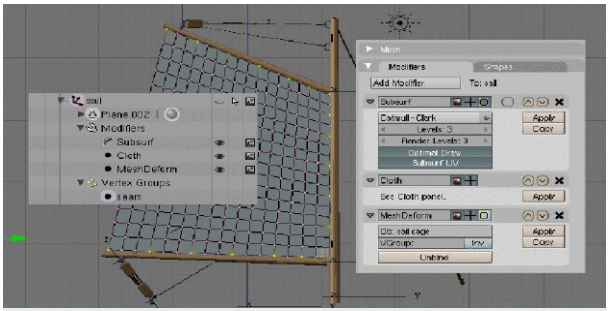


Figure 8: sail mesh, modifiers and vertex group

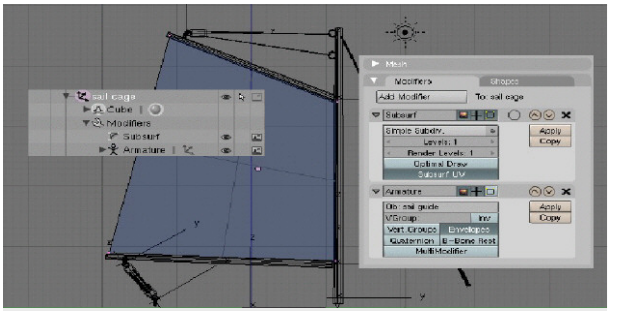


Figure 9: sail cage mesh with modifiers

Now we have the basic sail mesh. The subdivision was necessary a) to provide resolution for a nice sail deformation and b) to create a vertex group that will cause some nice wrinkles.

- To let the high-resolution sail mesh follow the boom movement, we will use a low-resolution mesh as a kind of cage together with the MeshDeform modifier.
- Create a cube object (Add››Mesh››Cube), name it "sail cage" and translate it to the sail position.
- Switch to EditMode and transform the "sail cage" cube mesh so that it encloses the sail mesh. Keep the space between cage and sail as small as possible.
- Now go to Buttons Window, Mesh Buttons, Modifiers Panel and add a Subsurf modifier to the sail cage. Use the Simple Subdiv type with Levels:1. This gives some flexibility for the cage.
- Now go to Buttons Window, Mesh Buttons, Modifiers and add as second modifier an Armature modifier with Ob:"sail guide". We can now use our "sail guide" armature to deform the "sail cage".
- Select the "sail" mesh, go to Buttons Window, Mesh Buttons, Modifiers and add two modifiers to the mesh:
- a) a Subsurf modifier, Type:Catmull-Clark with Level:2 or 3 to give more resolution to the cloth simulation. (room for experiments)
- b) a MeshDeform modifier with Ob:"sail cage". Press Bind to connect the cage to the sail mesh. The deformation of the "sail cage" will now translate to the "sail" mesh.
- Test the setup: select "sail guide", switch to Pose mode and rotate lower, middle, and upper bones. The booms should follow the armatures as well as the sail mesh.

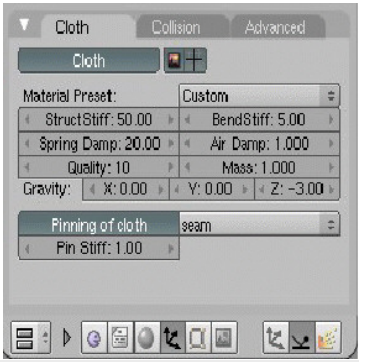


Figure 10: cloth parameters for sail mesh

by Johannes Ewers

by Johannes Ewers

Setup the cloth & wind simulation

We want to see some nice and realistic wrinkles on our sail. That could be done with a normal or displacement map. However, Blender provides a powerful wind machine we will use for our purpose. First, we have to give cloth properties to our sail.

- Move the "sail guide" to neutral position and select the sail mesh.
- Go to Button Windows, Object Panels, Physics Buttons, Cloth Panel.
- Click on Cloth and select Denim as cloth type (room for experiments)
- Click on Pinning of cloth and select the "seam" vertex group defined previously.

Now go to to Buttons Window, Mesh Buttons, Modifiers panel. You will see three modifiers top to bottom Subsurf, MeshDeform and Cloth. Click on the little arrow on the left side of the Cloth modifier to move it up one position in the stack. I had some problems with Cloth on the last position, sometime it works, and sometime the sail mesh is disconnected from its cage when running the simulation.

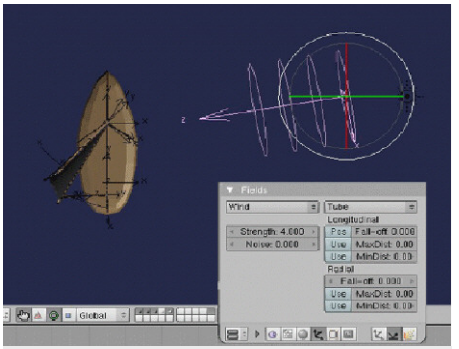


Figure 11: wind position and parameters

We are nearly finished. We just need some wind.

- Create an Empty and translate it to the side of the boat approximately two-boat length apart.
- Go to Button Windows, Object Panels, Physics Buttons, and Fields Panel.
- Choose Wind as Field Type; choose Tube as Fall-off. Set Strength: 4. Leave all other parameters at zero.
- Rotate the wind empty so that it points at the boat.

Ready to go!

Using the automation concept

Here comes the fun part. I provided a .blend file for experiments if you do not want to go through the modeling process.

- Select the "sail guide" armature and switch to Pose Mode.
- Rotate the lower bone to a 30-degree position (angle between boom axis and boat main axis).
- Rotate the upper bone to 40-degree.
- Rotate the middle bone to 45-degree. That gives the sail a realistic basic deformation.
- Go to the Timeline Window, skip to start frame 1 and start the simulation by pressing the play button. We need 30 to 50 frames to give the sail a nice deformation.

Watch Blender do its work. Some hints

Some hints

- Move the "sail cage" to a layer that you can deselect. It will react but does not obstruct the sight. Alternatively, use the restriction column in the Outliner (eye symbol).
- After a simulation, you have to clear the simulation cache before you can change the sail position. Select the sail mesh; go to Buttons Window, Object + Physics Buttons, Cloth panel, Collision tab. Press free cache.
- If you want to move the boat after a simulation, the sail will stay in place until you free the simulation cache.
- Do not get frustrated if the program behaves in unexpected ways.

Now add a top sail or a fore sail. Setup a second mast or model a more realistic hull. The result could look like my Bluenose schooner model. ■

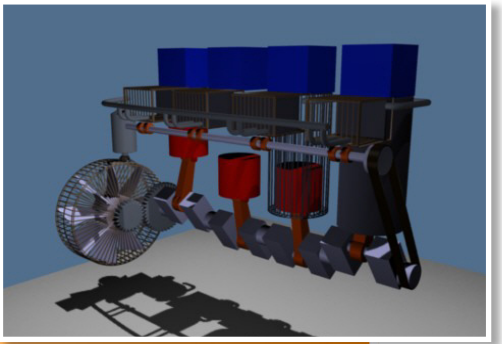
Johannes Ewers

I love the sea, old sailing ships, and maritime paintings. While I am not a good painter, I use 3D tools to create nautical scenes. After working with POV-Ray and Moray for some years, I switched to a combination of Vue d'Esprit and Blender.

You can find some of my images on my website, also at at Renderosity, and at Zazzle. .

Website: www.age-of-sail.com
e-mail: postbox1@age-of-sail.com

by Johannes Ewers



By William Edstrom

Introduction

For a while now I've had an idea for a free energy machine. Very similar in appearance to a standard four stroke engine, this new design uses magnetic pistons for power. Two pistons are pulled up by large permanent magnets, at which point a lead block interrupts the force, and the other two pistons, now in the power stroke, pull them back down. The action rotates a crankshaft, turning gears and an air compressor which gives the air pressure needed to move the lead blocks. A camshaft is pulled by a belt and times the air valves.

To see it in an animation, go here.

This article will show you how to rig the machine in a slightly better way than it was in this movie.

Some Basic Pointers

This article is designed for a wide range of Blender skills and experience, so don't feel insulted if I tell you a hotkey for something that you feel is newbie. I'm just trying not to lose anybody. That in mind, I'll try not to spoonfeed too much.

Remember while following the instructions to always place objects exactly. Use [Ctrl] or [Shift+S››Selection to Cursor] whenever moving an object and remain in orthographic mode[numpad5] and in straight on views[numpad1, numpad3]. It is useful to hide objects[H](unhide with [Alt+H]) and go into wireframe[Z] to see more clearly.

If you get lost because something new to you isn't explained well enough here, here are some links to cover the basics:

Basic Animation (setting Keyframes and working with the IPO window):

Advanced Animation in General.

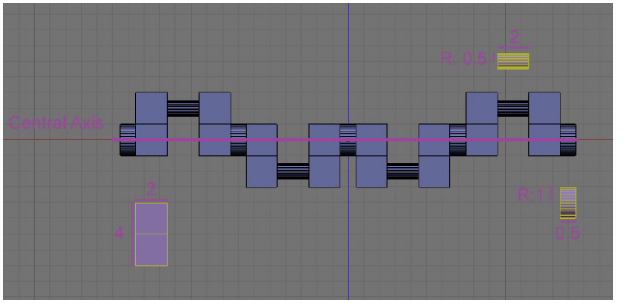
Constraints.

Modeling

This article is not about modeling, so I won't go through step by step instructions, but I will give an overview and measurements so you can model the perpetual motion machine.

The Crankshaft

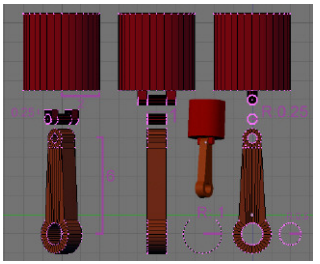
The crankshaft is just some cylinders and cubes. The two inside piston joints are a set and the two outside piston joints are a set. If you are going to rig your crankshaft with these instructions, the two inside should point down, and it should rotate around the X axis.



by William Edstrom

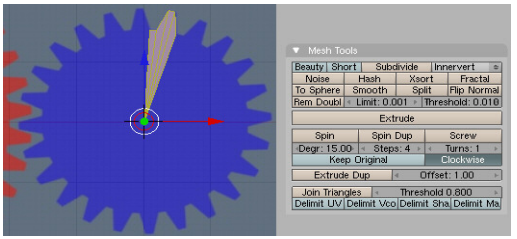
The Connecting Rod and Piston

The connecting rod and piston are two separate objects, but they are shown here together in edit mode.



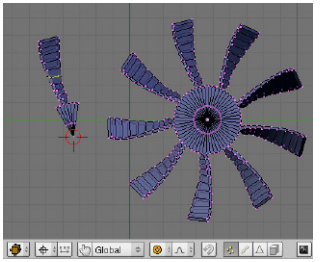
The Gears and Fan

To find the degrees of rotation for each tooth, divide 360 by the number of teeth. For this example there are 40 and 24. Spin Duplicate 360 degrees with the number of teeth as steps. Be sure to remove doubles[W›› Remove Doubles] and any internal edges.



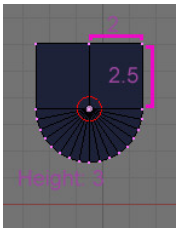
Download a useful background image here: (link to “Gears For Modelling(Not Shown).jpg”)

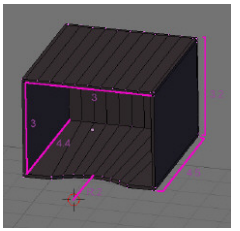
To give the fanblade it's curve, just rotate the middle vertices with proportional edit on.

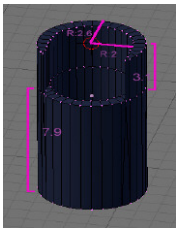


The Piston Cases and Lead

- The Lead Block
- The Lead Housing
- The Piston Housing

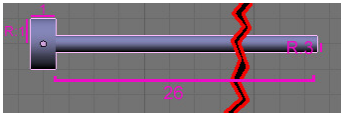


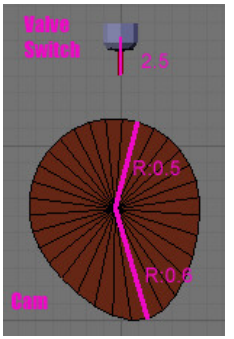


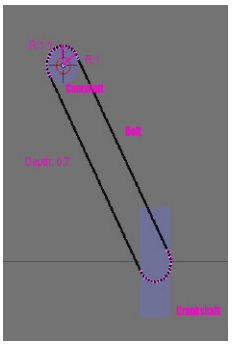


The Camshaft, Belt and valves

- The Camshaft
- The Cam and Valve
- The Belt

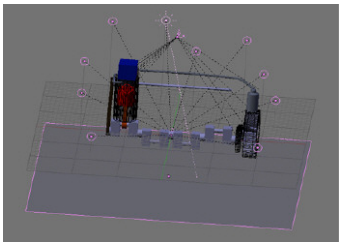






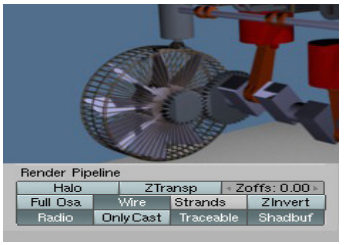
by William Edstrom

Lighting/Materials



Lighting in a mechanical demonstration can be very simple. In this scene I took a lamp, set it to 0.3 energy and made sure Rayshadow was off. Little lamps like these hardly affect render time at all, so you can copy eight or ten to light every corner and still have render times under five or ten seconds. A few seconds may not sound like a lot, but when doing an animation they really add up, so I like to do everything I can to shave seconds. This setup is very boring, however so I added a sun set to Rayshadow and Only Shadow. A plane underneath catches the shadow.

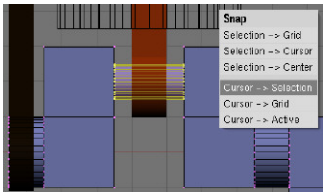
For the materials, mechanical drawings are also forgiving. Bright, solid primary colors – shunned like the plague almost everywhere else – are ok (just please turn down the specularity!), but should be left to the most important pieces and the most difficult to see. The stationary support structures are good as darker browns, grays and metallics. A simple cloud texture scaled very small makes a great normal map for nonmetal and dull metal parts.



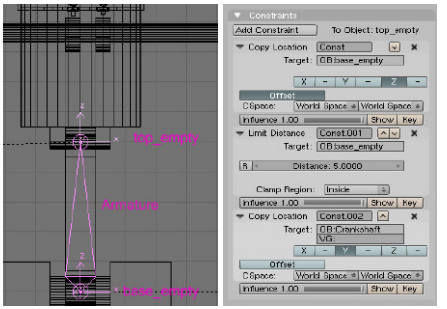
To show casing objects and their contents at the same time, one method is to render them in wire frame.

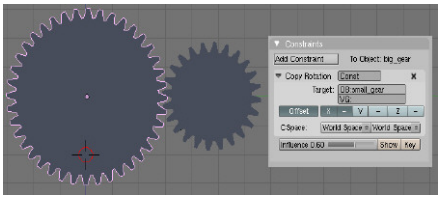
More complex and appealing setups than the ones described here can add life to a scene if it is to be more than a mere technical illustration. These are, however, outside the scope of this article.

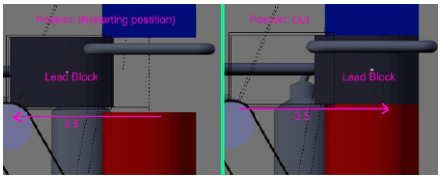
Rigging



The rigging setup mirrors, in a way, the mechanical processes in the machine. Rotation constraints act as gears and belts, location constraints like rods pushing back and forth. But because blender is primarily an animation suite, there are some tricks that have to be done differently than in an actual physical representation. The goal is to visually demonstrate, not build a physics simulation.







by William Edstrom

The Pistons

While doing the pistons, it is especially important to be thorough and follow instructions carefully. Even missing small steps can cause very strange looking reactions. Time spent double-checking will prevent headache in the long run.

Select the crankshaft and add a keyframe for rotation (I->Rot). Go ahead to frame 11 and rotate the crankshaft around the X axis 120 degrees. Key the rotation again and open the IPO window. Select the green X rotation keys and open the curve menu at the bottom of the window. Open Extend Mode and choose Extrapolation. Name the IPO “rotation”. By pressing Alt+A you can test the animation at any point to make sure all the steps up to that point work properly.

Next return to frame 1. All the rigging should be done in the neutral position in frame 1. Open the crankshaft mesh. Snap (SHIFT+S, cursor->selection) the cursor to the verticies of the first cylinder that will have a connecting rod. Copy a vertex and snap it to the cursor. Add an empty and name it “base_empty”. Select both base_empty and Crankshaft, then go into edit mode and parent “base_empty” to the vertex [Ctrl+P].

Snap the cursor to the piston object and add another empty there named “top_empty”. Go to the Objects Tab[F7] and add a Copy Location constraint. Set the target as “base_empty”. Press the offset button. This will make the object keep it's original distance from base_empty while still copying it's movements. Sometimes when the target is set the object will move, so use [Shift+S>>Selection to Cursor] to snap it back to the piston's center. Deselect the X and Y so the constraint only effects the Z location. Now is a good time to press Alt+A and pan around to get a good idea of what the constraints are doing.

Next, add a Limit Distance constraint and set it's target to “base_empty”. Set it to 5. This constraint keeps the piston within reach of the connecting rod. This next constraint will keep the limit distance constraint from moving the piston sideways. Add a Copy Location constraint and limit it to the Y axes, and set the target to “Crankshaft”. Alternatively, you could use a Limit Location constraint, but then you couldn't move the rig without breaking it.

Create an armature object at base_empty. In edit mode move the top of the first bone up 4 to the center of the piston. Go to Pose Mode and add a Lock Location constraint directly to “base_empty”. Still in Pose Mode add an IK Solver constraint with the target “top_empty”.

Select the piston and add a Copy Location constraint with the target “Armature”. Underneath the target a new input box for bones will appear. Type in “Bone” and set Head/Tail to 1.

In the Connecting Rod select all the vertices. In the Editing Tab(F9) create a new vertex group. Rename the group “Bone”. Press the Assign button to add the vertecies to the group. Parent the Connecting Rod to the armature and when it asks, choose Armature and Name Groups. Using an armature and IK solver to control the motion of the connecting rod is probably not as efficient as using a Track To constraint, but those are a major pain. Besides, this is a very simple way to learn about armatures.

The first piston set should be all done. To make sure everything is working before we duplicate it go to the side view and watch it in orthographic mode in wireframe. You should be able to see all the parts rotating in unison. You can parent a camera to the piston and in the edit tab set it to orthographic so you can watch it from there(set a new camera to main with

by William Edstrom

Ctrl+Numpad0). Any slipping should be noticeable from this view.

When duplicating, be careful to select all the parts at the same time(2 empties, connecting rod, piston, armature, you may need to hide[H] pars in the way) and [SHIFT+D] over to the next point on the crankshaft. Be careful to watch top_empty.001, as it likes to slide out of position at this step. Make sure it is resting precisely at the base of Piston.001 before proceeding. Next unparent "base_empty.001" with Alt+P and choose Clear and Keep Transformation. Then parent it to a new vertex as described earlier. Repeat until all four pistons are working.

The Gears

The gears must turn together, in opposite directions. Because they are different sizes they must also turn at a different speed. This difference can be found by the ratio of teeth. These particular gears have 24 and 40 teeth, giving a ratio of 5/3. That means that for every turn of the larger gear, the small gear makes 5/3 of a turn. For simplicity of animation we are going to animate the smaller gear and use it to turn the larger at 3/5 it's speed. This can be done with a single constraint.

Select the little gear, and in the IPO window choose the curve "rotation" (If you have skipped ahead to this point to read about gears, this ipo will not exist. Create it by following the steps in paragraph 2 of pistons). Add a copy rotation to the larger gear, targeting "small_gear". Deselect Y and Z, then push the negative sign next to X to invert the motion. Now the gears will turn in the opposite direction. To slow the larger gear, set the influence to 3/5, or 0.6. This is the reason that even though the larger gear is actually providing power, we use the small gear. Influence does not allow inputs of greater than one. Animate the small gear the same as the crankshaft. Now they should look just about right, although you may need to rotate the larger gear so that it appears to make proper contact(press offset to allow the rotations to stick). It's hard to tell that the gears aren't going exactly the same speed as the crankshaft, so in this animation we'll leave them separate, because it makes timing everything easier(who doesn't hate fractions?). If it was necessary to keep them going at precisely the same time we would simply constrain the crankshaft to follow the large gear and do the math for timing the other parts.

The Lead Blocks

The lead blocks should be covering the piston housing on the downstroke and be fully retracted during the upstroke, moving between in three frames.

Make three copies[Shift+D] of the lead block, each moved 7 along the X axis. If you like, you can also copy the stationary parts also.

Select the first lead block. It will need five keyframes to define the motion.

Frame Position

29 In

31 out

44 Out

46 In

59 In

In the IPO window, select the X location curve and open Curve››Extend Mode››Cyclic Extrapolation. Now this lead should cover the top of the piston housing during a downstroke and uncover the housing during an upstroke. Name the IPO curve “lead_outside”. Delete the Loc X keys in the IPO window. Add lead outside to the other outside lead.

Select a middle lead block. Since the middle pistons are on the opposite stroke, they need to be timed differently. Push the number next to the IPO title so that changes made to this one won't affect the others. Rename it “inside_lead”. Move the motion curves back 15 so that the first keyframe is at 14. Add “lead_inside” to the other middle block. The lead blocks should now be synchronized with the rest of the machine.

The Valves and Camshaft

The camshaft opens air valves at the right time to move the lead blocks. The shaft itself copies the rotation directly from the crankshaft and the cams copy their rotation from the shaft. The valves are already timed to go at the right time, you just need to push offset and rotate the cams so they appear to be pushing them.

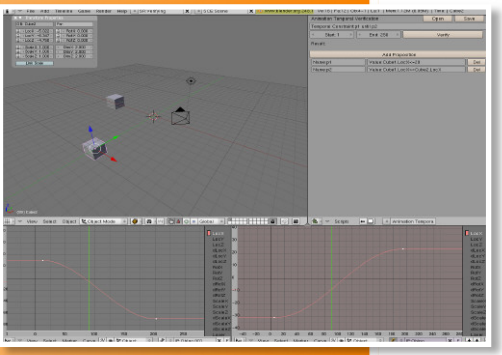
If you modeled it yourself, the valves obviously won't already be timed for you, so you will have to do it yourself. Just rotate the cams so they hit they're valves just before the block moves, one cam for each direction. Then key the valves so they appear to move.

Application

The rigging setups used here can be used for a lot more than my perpetual motion machine. A four stroke engine is a given, as that was the inspiration, but there are also many more uses with very similar rigs. Even character animation relies on many of the same principles described here.

And hopefully, if you have some crazy mechanical contraption that has been locked away in your noggin, the examples shown here can help bring it out. ■

by William Edstrom



By Alain Mathez

Introduction

Like a lot of people, I have plenty of 3D ideas in the head, and most of the time I begin by directly drawing something in Blender. And like a lot of people, I know this is not a really good method, but I do all the same because I am too lazy to follow a better way.

Two weeks ago, I was on a wellness week-end with my girl friend. Three days without any computer, hell or paradise? Well, without computer, but not really without Blender, because I could not get some of the ideas out of my mind. So I just had to relax my body and let my brain choose between “no-thinking” and “B…2.48a”. Guess what, I took a pen and a paper, and began to draw some sketches.The best part in that story is that I had a lot of time to grind the project, before I could finally start to build some objects in Blender. The ideas where numerous and growing every minute; I had already a list of problems and some solutions and many derivations and alternatives.

Once back at home, I tried to do some critical points in Blender, and I submitted the unsolvable knots to the legendary famous BlenderClan.

***Note for the beginners (like me, sometimes)** : try to reduce your problem to its simplest state to find solutions. Open a new Blender file, with just the things you need to tackle the problem, and experiment with it. If you are unable to find a solution, you can ask about it at blenderartists forum, explain the case and maybe give them the file. If you are a bit lucky, you will receive a lot of suggestions. Be careful not to ask too many things at one time, the answers will probably be lesser.*

Definitions

Here you can see some fundamental definitions for understanding what we can do with this tool to improve the time of checks and verifications.

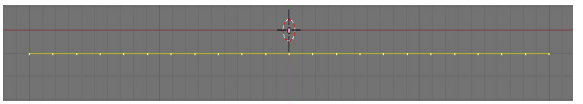
Well, enough said, let’s get on with the project. The main criteria are:

- Doing animations with only one IPO (The "One Ipo Research Attitude" : OIRA) … lol, this name just make me laugh myself
- When not possible, than "As Few As Possible Ipo Research Attitude" : AFAPIRA)
- In the “OIRA or AFAPIRA” serial, we are starting with : Doing the wheels of a vehicle staying on the floor (Mesh) with constraints Doing a Caterpillar soft Track, with dynamic inertia No addition of Python scripting, I am not yet a experienced in it (PyDrivers are allowed)

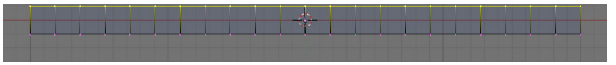
Use Modifiers and Constraints from Blender 2.48a

1. Layer 1 : the floor

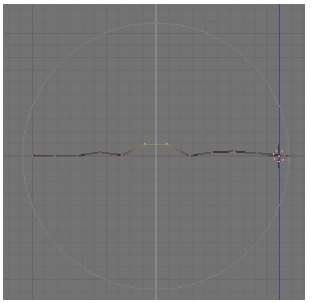
Open a new Blender File, delete all objects, select only Layer 1 (1), switch to Top View (Numpad 7), add a Mesh Plane (Space, Mesh, Plane), in Edit Mode (Tab) select (RMB, Shift+RMB) and delete (X, 1) the 2 upper vertices, select the remaining Edge (A) scale it 20x (S, 20, Enter), subdivide it 21x (W, 2, Number of cuts :21).



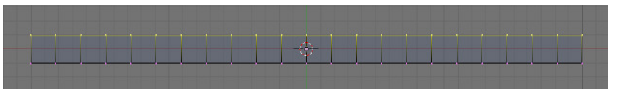
Select the vertices on the positive part of the X axis (B, LMB+MouseMove), and hide them (H).



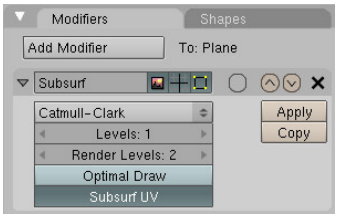
Switch to Front View (Numpad 1) select the two middle vertices, switch to the Proportional Edit Tool (O), select Random Falloff, move the vertices 1BU along the Z axis, having all the vertices in the select circle (G, Z, Mouse-wheel)

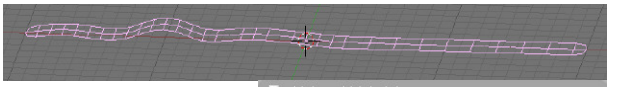


Switch to Top view (Numpad 7), reveal the invisible vertices (Alt+H), select all the vertices (A, A), extrude 2BU along the Y axis (E, 1, Y, 2, Enter)

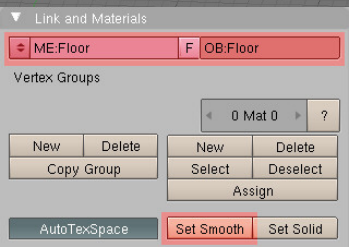


Select all the vertices (A, A), recalculate normals outside (Ctrl+N), swtich to Object Mode (Tab), add a Modifier Subsurf (Ctrl+2).



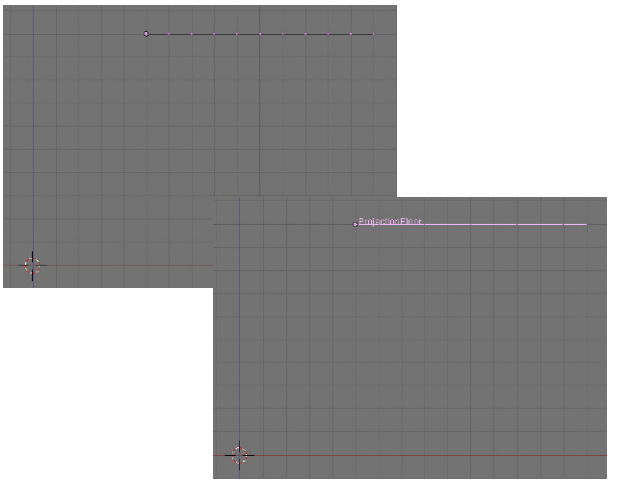


Rename the Plane in Floor and make it smooth.



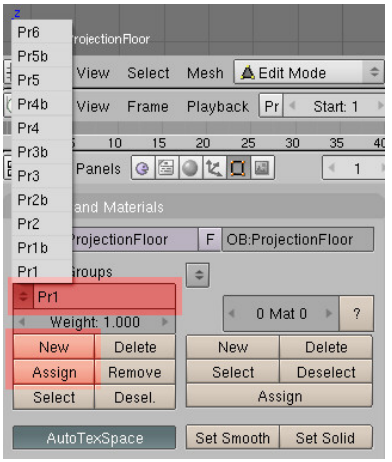
2.Layer 2 : The projections

Select only Layer 2 (2), switch to Top View, add a Mesh Plane and modify it to obtain 10 Edges from 1BU(Blender Unit) on a straight line, place it on X=5, Y=0, Z=10, give it the name ProjectionFloor and watch of the center on the object !

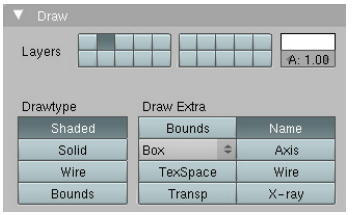


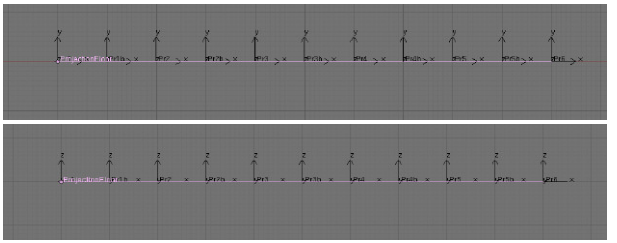
by Alain Mathez

In Edit Mode, go to the panel "Link and Material" (F9), add a Vertex Group (New), name it Pr1, select only the first Vertex on the left, press Assign. Add a Vertex Group (New), name it Pr1b, assign to this Group only the second Vertex, then make the same for Pr2, Pr2b, Pr3, Pr3b, … Pr6.



Add an Empty, snap it to the grid on the same position as the second Vertex, name it Pr1b. Go on with Pr2, Pr2b, … Pr6. Verify in Top View that the position of the Empties is correct on the Y axis too. In the panel "Draw" (F7), show the name of the different objects on Layers 2 and 3.

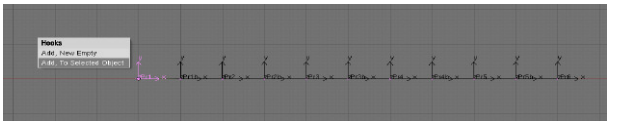




3.Layer 3 : the Hooks

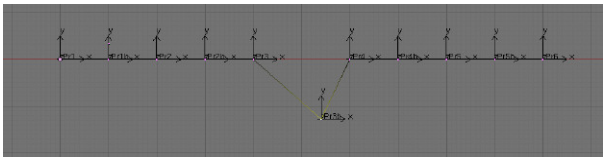
Keeping Layer 2 active, select Layer 3 (Shift+3), switch to Front View, add an Empty (Space, Empty), name it Pr1, place it exactly on the same place as the first Vertex from ProjectionFloor, using the Ctrl Key to snap to the grid (G, Ctrl+MouseMove). If the Empty doesn't have his Z axis pointing up, check in the User Preferences that the button "Aligned to View" is not enabled.

Select the Mesh ProjectionFloor (RMB), switch to Edit Mode (Tab), select only the first Vertex on the left (RMB). Staying in Edit Mode, select the corresponding Empty Pr1 (Ctrl+RMB) and create a Hook on the Vertex (Ctrl+H, 2).

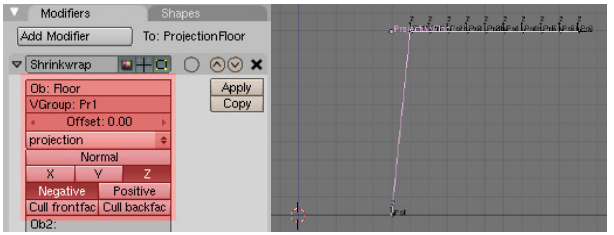


by Alain Mathez

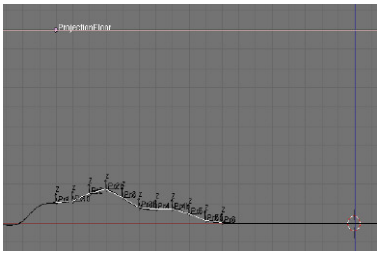
Then select the second Vertex (RMB) and the second Empty (Ctrl+RMB), create the Hook (Ctrl+H, 2), and so on to the last Empty. Now when you are finished, if you select one Vertex and move it (G+MouseMove), the corresponding Empty will follow.



Well, that was just to be sure that the setup is working, so don't move any Vertex for the moment (Esc). Now the first trick. switch to Front View (Numpad1), show the first Layer (Shift+1) to see the Mesh Floor, select the Mesh ProjectionFloor, go to the panel "Modifiers", add a Modifier Shrinkwrap, set the parameters and enjoy the result.

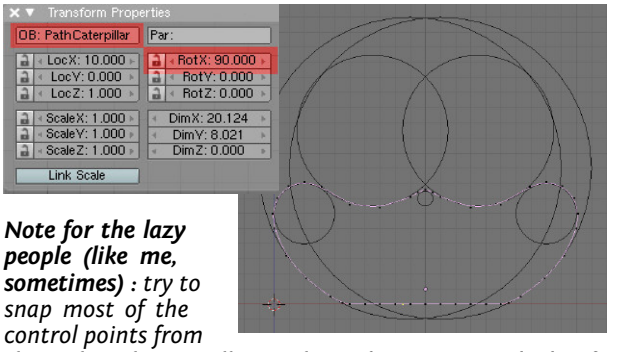


Guess what, you have to do the same for all the other Vertex Groups, ... but don't panic, when you click on the button "Copy" in the Modifier, you get to copy all the attributes automatically, the only change you have to make is to the VGroup:Pr1b, Pr2, Pr2b, ...Pr6). Now if you move the Mesh , you can see the Empties following the Floor. Pretty cool, isn't it ?

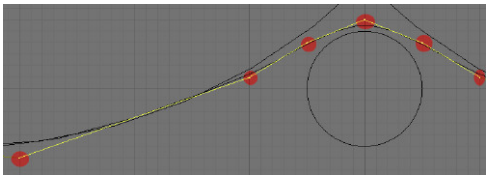


4.Layer 4 : the Path

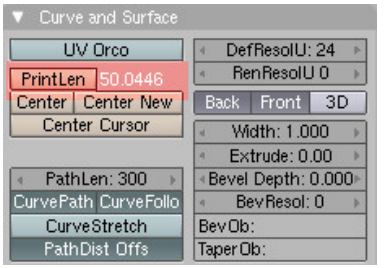
Show only the Layer 4 (4), Front View (Numpad1) add some Curves Bezier as landmarks to position the Curve NURBS Circle which will be the Path of the caterpillar track. Because we don't need 3D Curves, it is necessary to turn the Curves around the X axis. Show the panel "Transform Properties" set the rotation for RotX. And don't forget to give names to all your objects.



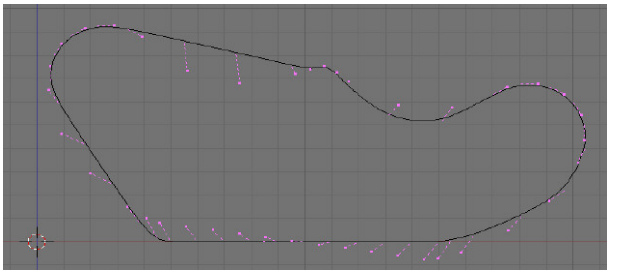
***Note for the lazy people (like me, sometimes)** : try to snap most of the control points from the Path PathCaterpillar on the grid (zoom around a bit if necessary) because soon you will have to align Bones and Empties on every control point of the path. So it is easy to do it with an extreme precision where every object has his place on the grid.*



by Alain Mathez

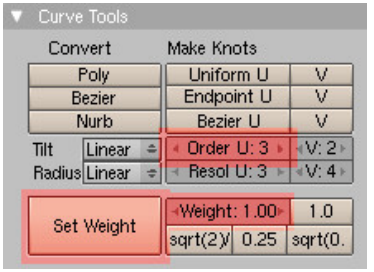


Note for everyone (like me, sometimes) *: ensure that you have a round length from your path when you've finished designing it. For example, 50, because the caterpillar track will be easier to configure with (for example) 50 links, having every 1BU width.*



The path must look like a slack rope now because it will look like a tight rope soon.

Select all the vertices, go to the panel "Curve Tools" (F9 when the Curve is in Edit Mode), set the Order U to 3, and attribute to all the vertices a weight of 1.

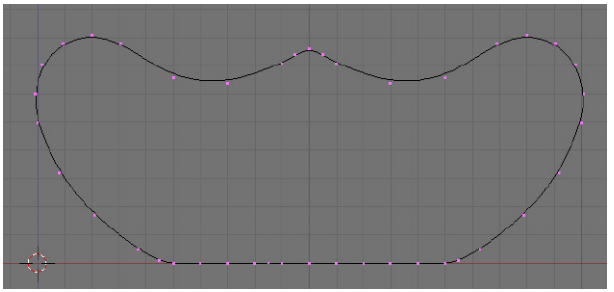


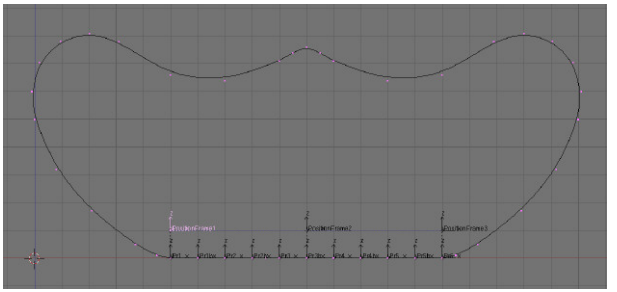
Note for the pressed people (like me, sometimes) *: it is important to draw the Curve PathCaterpillar like a slack rope, because when the caterpillar moves, the track will tighten. But remember, it's not elastic, so the length must be kept constant.*

5.Layer 5 : the Constraints

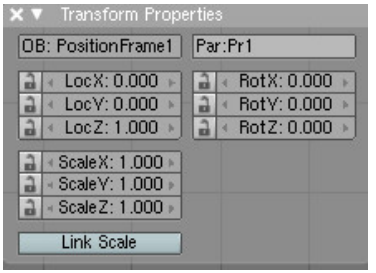
We are going to configure a lot of constrained Empties. This system will help us to position the steel frame of the vehicle, and to deform the future Armature which will itself control the Curve PathCaterpillar.

Show Layer 3, 4 and 5 (3, Shift+4; Shift+5), add an Empty, name it PositionFrame1, set its size as 0.5 in the panel "Links and Material", give it the Empty Pr1 as Parent (RMB on PositionFrame1, Shift+RMB on Pr1, Ctrl+P, Enter), and move it 1BU above Pr1. Do the same with PositionFrame2 (Parent = Pr3b), and PositionFrame3 (Parent = Pr6).

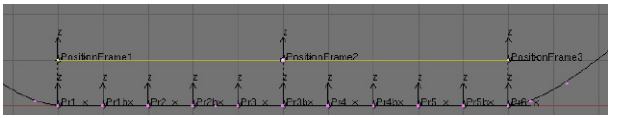


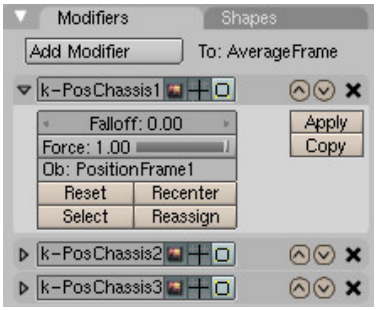


by Alain Mathez

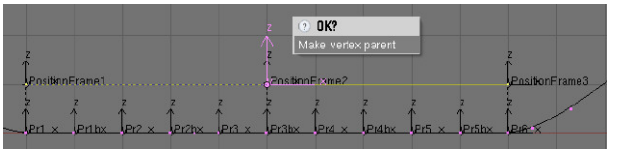


Add a Mesh Plane, name it AverageFrame, modify it to obtain 3 vertices on a straight line, with every vertex on the position of PositionFrame1, PositionFrame2 and PositionFrame3. Hook each Vertex to the corresponding Empty.

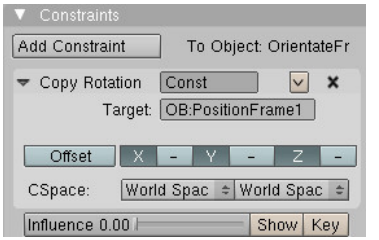


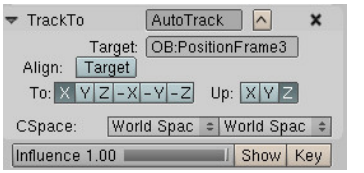


Switch to Object Mode, add an Empty, name it OrientateFrame, place it on PositionFrame2, select the Empty OrientateFrame (RMB) and the Mesh AverageFrame (Shift+RMB), switch to Edit Mode (Tab), select all the 3 Vertices (A), and define the vertices as Parent from the Empty (Ctrl+P, Enter).

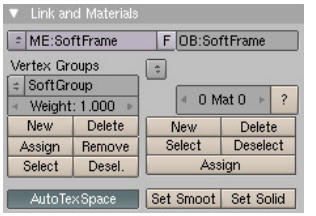


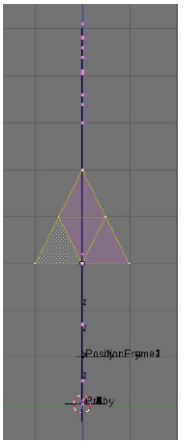
Now if you show the Layer 2 (Shift+2), select the Mesh ProjectionFloor, and move it along the X axis, you can see that the Empty OrientateFrame follows the Mesh AverageFrame, but where the Mesh Floor in Layer 1 (hided) goes up, OrientateFrame goes up side down, having the Z axis pointing downwards. To correct this, simply add it as a Constraint Copy Rotation on PositionFrame1, Influence 0, and a TrackTo (Ctrl+T, 1) on PositionFrame3. So you have 2 choices to test later, just play with Influences.



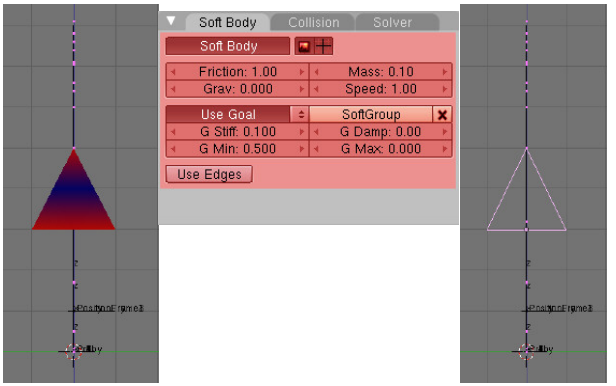


Switch to Side View (Numpad 3), add a Mesh Plane, name it SoftFrame, go in Edit Mode (Tab), and rotate the Plane around the Y axis (R, Y, 90, Enter). Select only the 2 upper vertices, merge them (W, 5, 3, Enter), select all (A, A), subdivide (W, 1), and move all 1BU up (G, Z, 1, Enter). Set the parameters in the panel "Soft Body" (F7, F7)



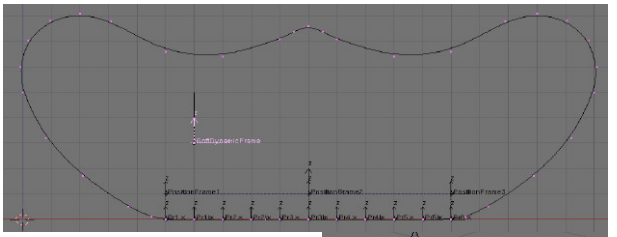


by Alain Mathez

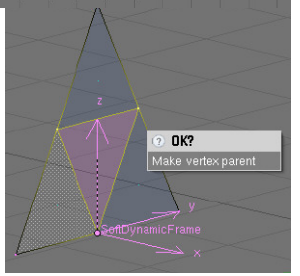


This method is not mine, you can read about it here And you will find the full description here Thank you yoyofargo!

Add an Empty at the origin from SoftFrame, name it

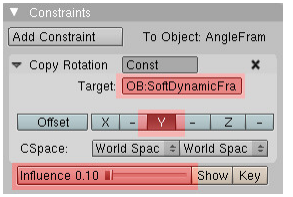


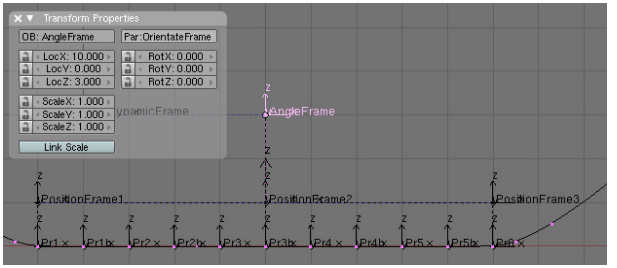
SoftDynamicFrame, and give it the 3 central vertices from the Mesh as Parent.



So, when the SoftBody will move, the Empty SoftDynamicFrame will receive an angle of the movement. This angle will be copied to move parts from the future Armature.

Add an Empty AngleFrame 2BU above OrientateFrame, define OrientateFrame as the Parent of AngleFrame. Add a Constraint Copy Rotation from SoftDynamicFrame to AngleFrame, leave only Y and Infuence 0.1.

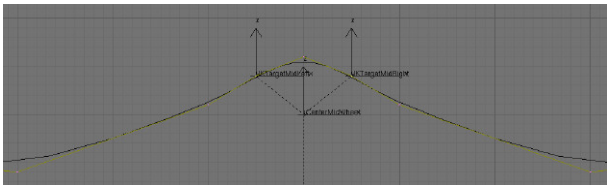




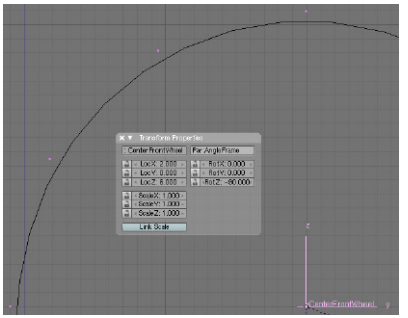
Select the Curve PathCaterpillar, and give it the AngleFrame as Parent. Now if you move the Mesh ProjectionFloor, PathCaterpillar will follow the slope of the Floor. If you rotate the Empty SoftDynamicFrame around Y axis, PathCaterpillar will follow the rotation.

Hey, grab a drink, it's not finished yet. Add an Empty CenterMidWheel, child of AngleFrame, add an Empty IKTargetMidLeft and an Empty IKTargetMidRight both child of CenterMidWheel. IKTargetMidLeft and IKTargetMidRight must be placed on the corresponding vertices from PathCaterpillar.

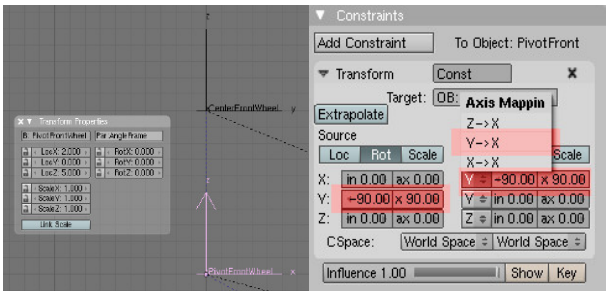
by Alain Mathez



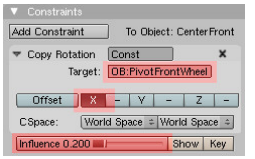
Add an Empty CenterFrontWheel, child of AngleFrame. Rotate it -90° around Z axis (R, Z, -90, Enter). This special orientation will be useful for a future IK constraint.



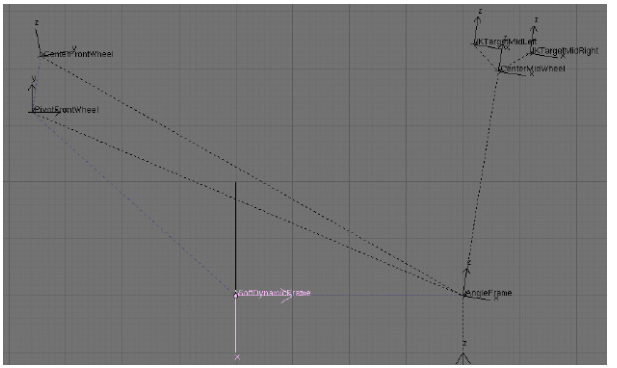
This Empty will have to turn in the other way round of AngleFrame. To do the magical trick, we need another Empty PivotFrontWheel, child of AngleFrame, and a Constraint Transformation on PivotFrontWheel. The Target Object of the Constraint is SoftDynamicFrame.



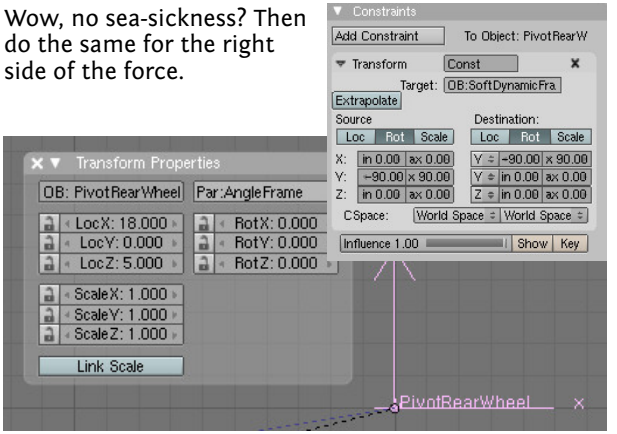
CenterFrontWheel will receive a Copy Rotation Constraint from PivotFrontWheel.



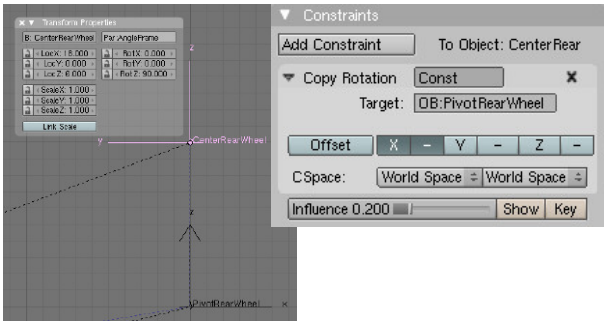
Now if you rotate SoftDynamicFrame, you should be able to see something like this.



Wow, no sea-sickness? Then do the same for the right side of the force.

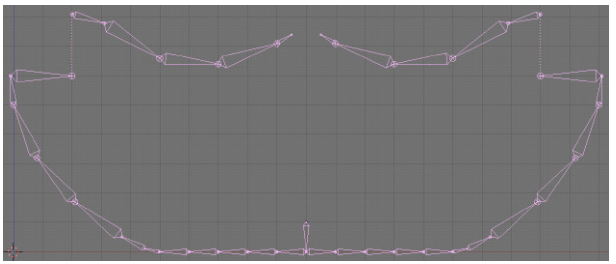


by Alain Mathez



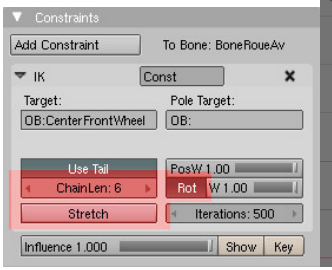
6.Layer 5 : the Armature

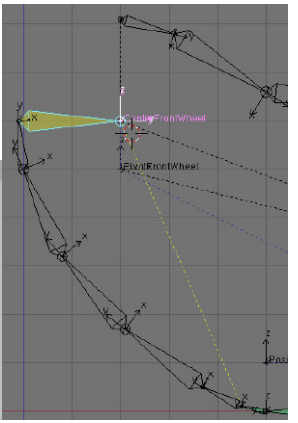
Add an Armature on Layer 6. Remember that each Bone starts and ends on a Vertex from the Curve PathCaterpillar (you might see some exceptions on the picture below). Origin of the Armature is the position of OrientateFrame, and Parent of the Armature is OrientateFrame too.



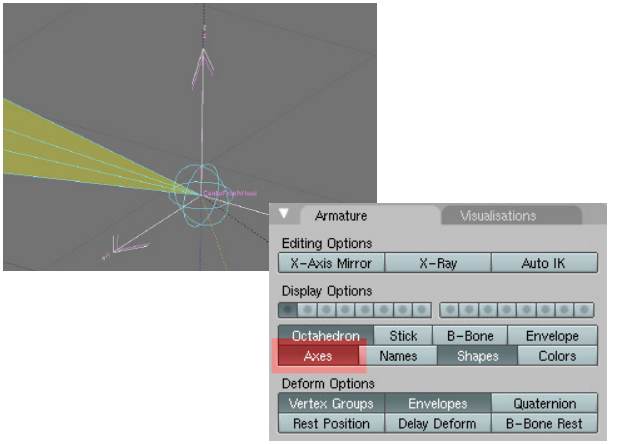
When the Armature is designed, in Edit Mode select all the Bones (A) and Recalculate Bone Roll Angles (Ctrl+N, 1).

It's time to add IK Constraints. Armature is in Object Mode. Select the Empty CenterFrontWheel (RMB), select the Armature (Shift+RMB), switch Armature to Pose Mode (Ctrl+Tab), select the Bone which already pointing on the Empty, add the Constraint to active object (Shift+I, 1)



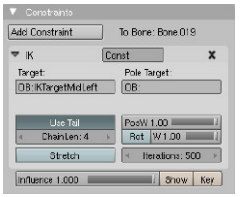


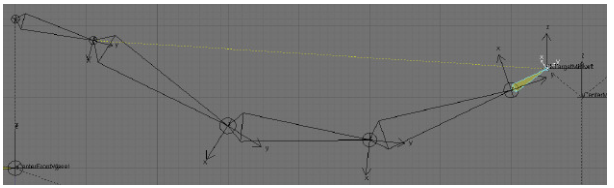
Don't forget to disable the Stretch button, we need a constant length. The option Rot lets the Bone follow the rotation of the target, and you, lucky people, have already set the orientation of the target CenterFrontWheel in part 5, didn't you ?



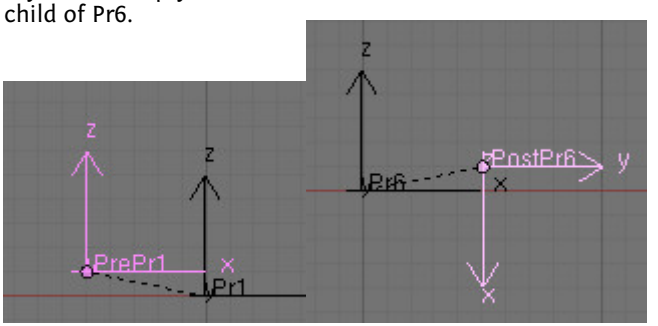
by Alain Mathez

What else? The IK for the middle wheel.

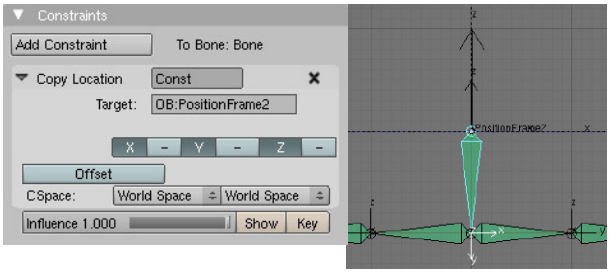


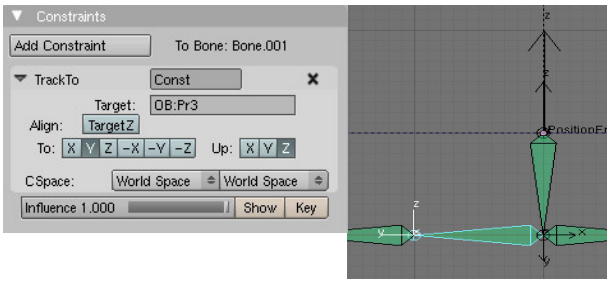


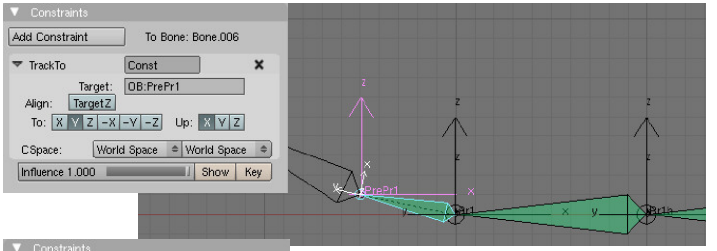
Have you twigged? Ah, by the way, you must add in Layer 3 an Empty PrePr1, child of Pr1, and a PostPr6, child of Pr6.

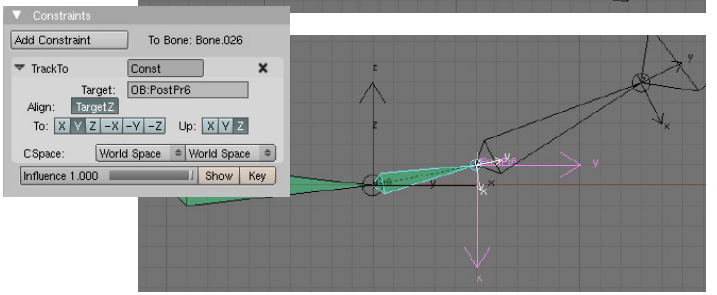


And the last, but not least, the constraints for the floor.



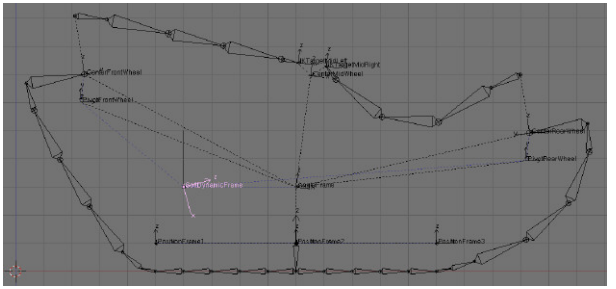






by Alain Mathez

Show the Layers 5 and 6, rotate the Empty SoftDynamicFrame (+ and -90°), what do you see? Armature is deformed but not stretched? Goal reached! Well, in fact not completely. The IK for the middle wheel is only effective when it stretches a little bit.

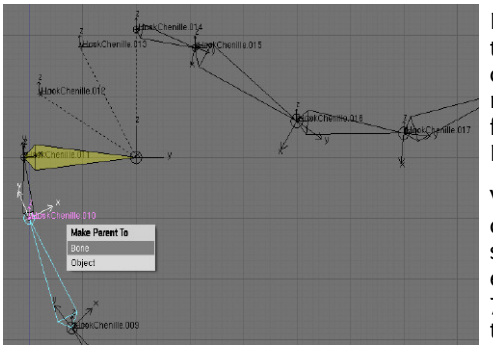


***Note for the experts (like me, maybe one day)** : if you find a better way to use the IK Constraint, without any stretching, with a full constant length for the complete chain, please share some tips as how you achieved it!*

7.Layer 7 : the Hooks

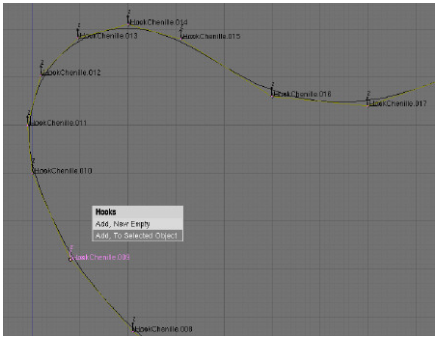
This one is only for pleasure. On each Vertex of the Curve PathCaterpillar place an Empty. Names are HookPath... HookPath.0XX. Well, on the picture below the names are still in French, but it's too late to translate this in detail, and I'm not ready with the extra work, so please translate for me HookChenille = HookPath.

Select the Armature go in Pose Mode (Ctrl+Tab), select an Empty (Ctrl+RMB), select the corresponding Bone (Shift+RMB), make the Bone Parent from the Empty (Ctrl+P, 1)



Do I have to be precise, it's necessary for each Empty?

When it's done, show Layers 3 and 7, select the Curve PathCaterpillar, go in Edit Mode, select 1 Vertex (RMB), select the corresponding Empty (Ctrl+RMB), make the Hook (Ctrl+H, 2).

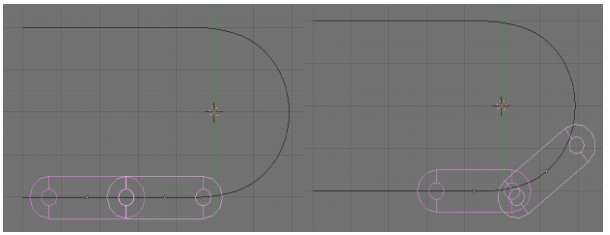


And one more time, I'm afraid that's necessary for each Empty. But if you're a good pianist, it doesn't take you a long time to perform.

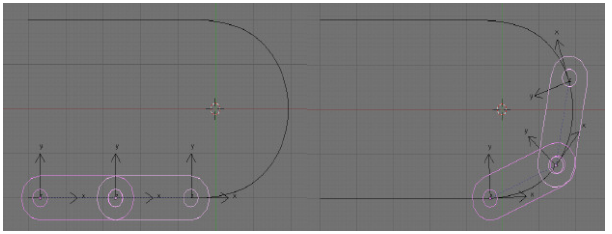
Now we start a special step: making the track links following the path. This will be always be only an approximation. Why? Because I know of only two methods and both are not "real".

First method: origin of the links are on the path, then the joints are too distant on the little curves.

by Alain Mathez



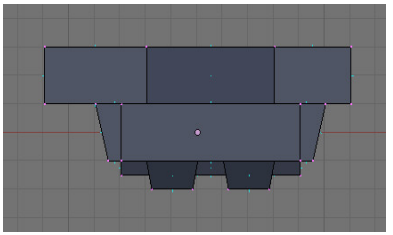
Second Method: origins of the links are working with a TrackTo Constraint, but the joints are too close on the little curves.

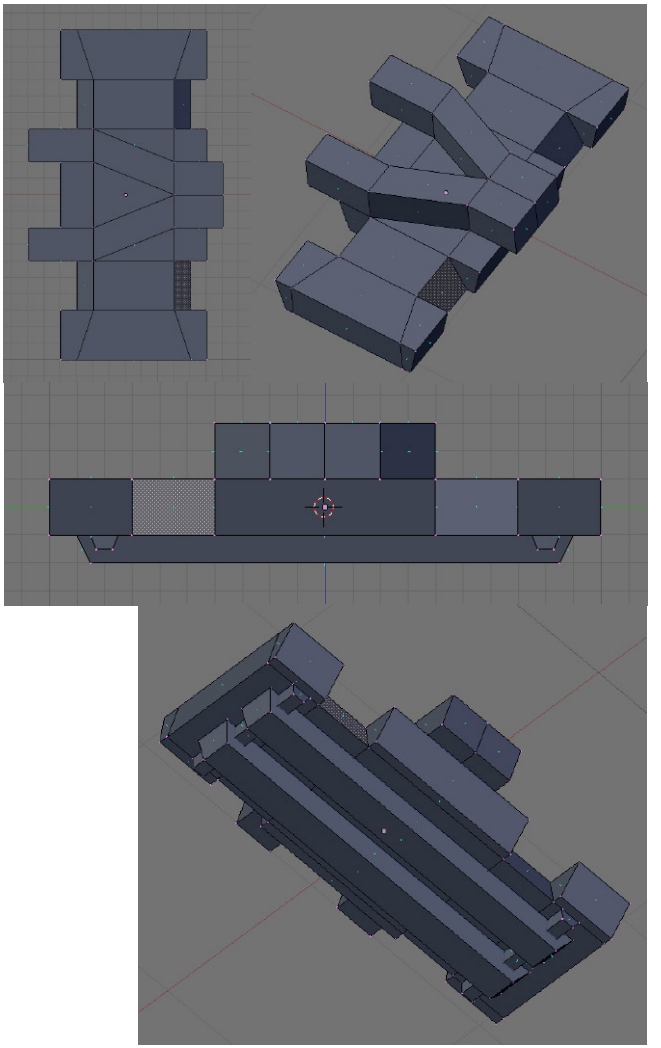


Well, first method seams to be more imprecise, but faster to set up. We'll take this one for now, with narrow track links. The illusion should be enough.

8.Layer 8 : the links

Model a simple link about 1BU (X axis), 2BU (Y axis) and 0.5BU (Z axis). You could do it with a deformed Cube, or like me, a bit more complex.

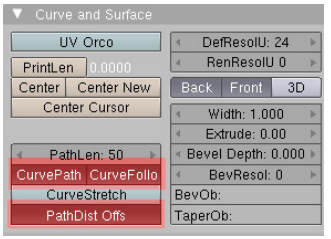


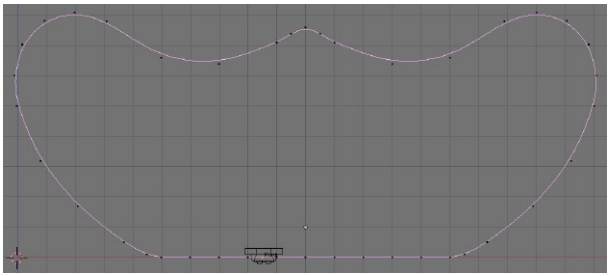


by Alain Mathez

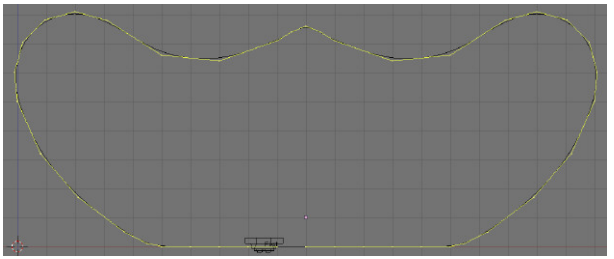
by Alain Mathez

Select the link (RMB) select the Curve Path-Caterpillar (Shift+RMB), make the link follow the Curve (Ctrl+P, 2). Clear the rotation (Alt+R, Enter) and the origin (Alt+O, Enter) of the link. If necessary, re-orientate the (R) Link to align it along the Curve. Make sure that the Curve is configured with “CurvePath”, “CurveFollow” and “PathDist Offs” in the panel “Curve and Surface”.



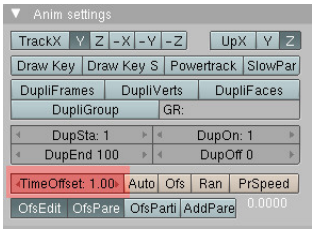


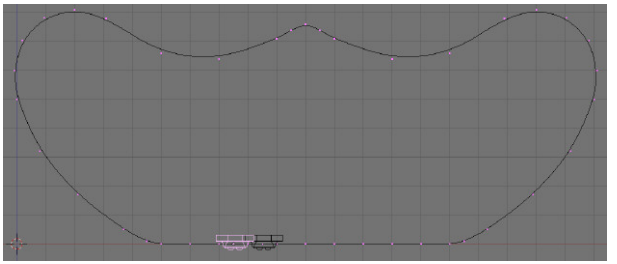
If the link is not on the straight portion of the Curve, select the Curve, go in Edit Mode, select all. The only non-yellow edge (not selected) is probably not in the middle of the straight portion.



Select one Edge in the middle bottom, erase it (X, 2), the problem should be solved. If the Curve is open, close it (C).

Quit the Edit Mode of the Curve. Select the link, make a duplicate (Alt+T), give it a TimeOffset to 1.00 in the panel “Anim settings” (F7)

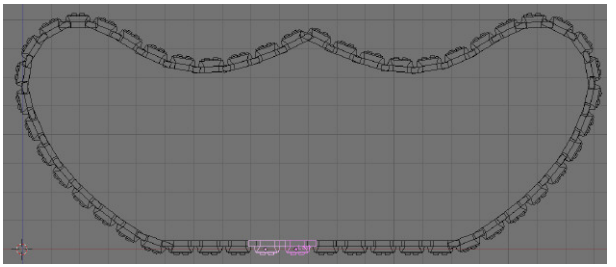




With the duplicate method (Alt+A and not Shift+A) you can modify a link, selecting each duplicate and modifying at the same time. So you do your test with a simple deformed cube, and after the 50 links are ready, you can tweak their appearance without doing the work 50 times.

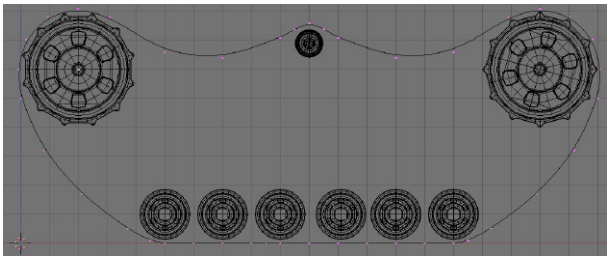
Note for the smart people (like me sometimes, but not enough for this case) *: there is maybe a magic trick with an Array Modifier or something like this to add all the links with just as few clicks. If you find it, please let me know.*

You remember the round length of the Curve Path-Caterpillar? It was about 50, but the link is about 1BU width, so you have to do the same again 48 times, Alt+A, TimeOffset: 2.00, Alt+A, TimeOffset: 3.00, ...

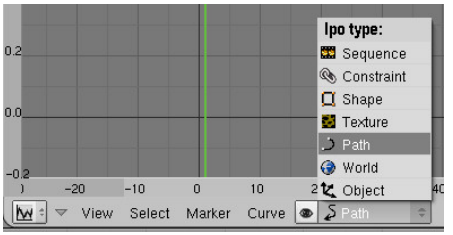


9.Layer 9 : the wheels

This part is left as an exercise for the reader. I will just show you my Layer as example. I began with the first wheel, upper left, and the others are modified and scaled copies.

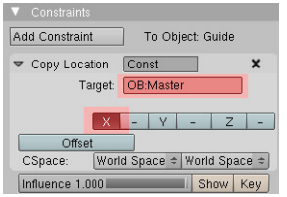


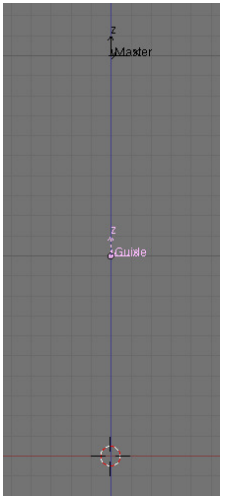
Parent of the wheels are on the Layers 3 and 5. FrontWheel is a child of Center-FrontWheel, MiddleWheel is a child of CenterMidWheel, RearWheel is a child of CenterRearWheel, FloorWheel1 is a child of Pr1, FloorWheel2 is a child of Pr2, ..., FloorWheel6 is a child of Pr6.



10.Layer 10 : the Master and the Guide

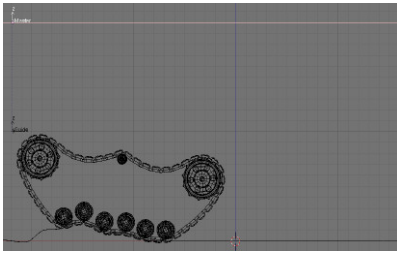
On Layer 10 add two Empties. The first one, Master, must be placed at 0 on the X axis, 0 on the Y axis, and 20 on the Z axis. The second, Guide, must be placed at 0 on the X axis, 0 on the Y axis, and 10 on the Z axis. Guide will then receive a Copy Location Constraint from Master, only for the X-axis.





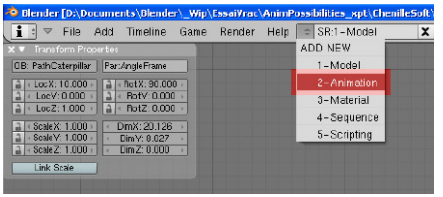
Guide is the Parent of the Mesh ProjectionFloor on Layer 2, and Guide is the Parent of the Mesh SoftFrame on Layer 5.

Now if you move the Empty Master along the X axis and above the Mesh Floor, you should see the ensemble following the surface of the Floor.

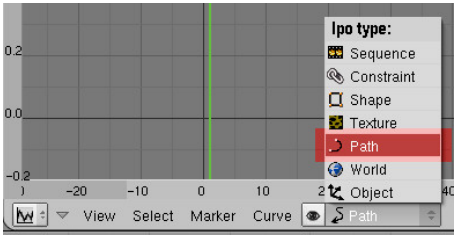


by Alain Mathez

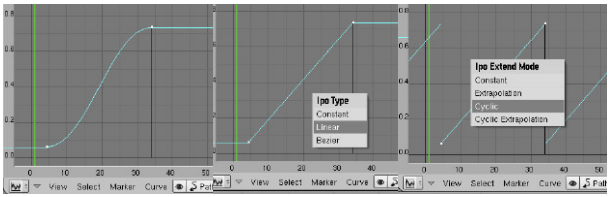
Is sliding not enough? Don't give up, you've already done the greatest part. Go to the "Animation" Screen Layout (Ctrl+?) or from the menu.



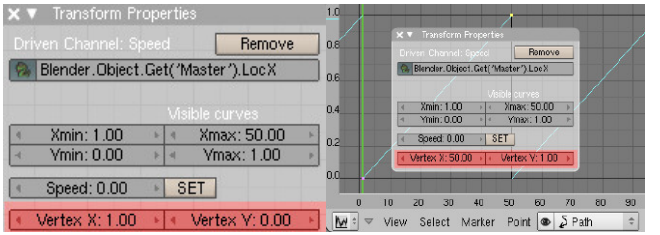
Select the Curve Path-Caterpillar (on Layer 4), in the window "Ipo Curve Editor" select the Ipo Type to Path.



If there is no Ipo Curve in the window, add a Vertex near from the origin of the axis, (Ctrl+LMB) and another a bit further (Ctrl+LMB). Specify the Interpolation Mode as Linear (T, 2), and the Extend Mode as Cyclic (E, 4).



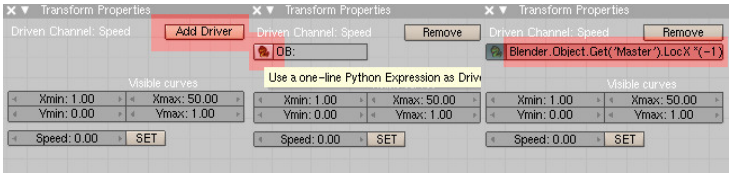
Open the "Transform Properties" box, edit the Ipo Curve (Tab), select the first Vertex (RMB) give it the coordinates in the box.



The second vertex has X coordinate 50 because this is the length of the Curve PathCaterpillar. If you don't choose Cyclic for the Ipo Curve, the links will do only one lap along the Path.

Now the mysterious miraculous PyDriver. Write "Blender.Object.Get('Master').LocX *(-1)"

Tadaaam!



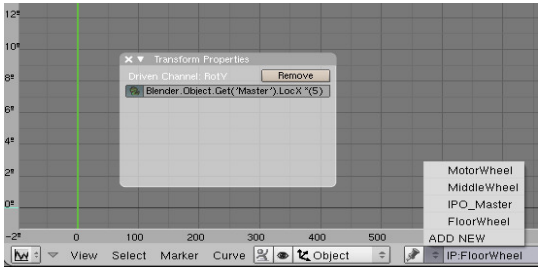
If the links move in the wrong direction, you can inverse the Ipo Curve (like "\ \ \" and not "/ / /"), or add a *(-1) factor at the end of the PyDriver.

Showing only the Layers 8 and 10, select the Empty Master and move it along the X axis. Impressive? Wait to see the wheels! Select the FrontWheel, select the Ipo Type Object, Select RotY (LMB), click on the "Add Driver" button in the "Transform Properties" box, click on the snake and write, "Blender.Object.Get('Master').LocX *(3.05)".

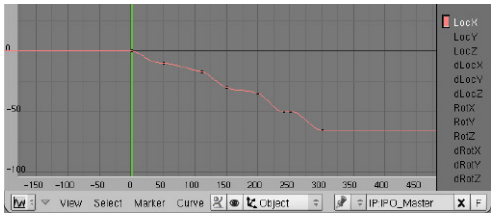
by Alain Mathez

Note for the mathematician people (not like me, I'm afraid) *: I have absolutely no idea why the factor is 3.05, but I find it good in testing. It depends naturally on the teeth on the wheel, but in my case 50 / 12 = 4.16. So, who knows why?*

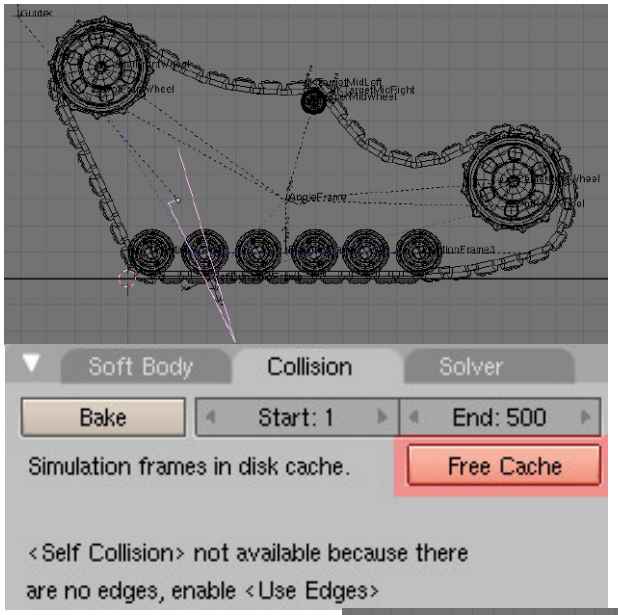
Apply the same tweak for the other wheels. You can of course re-use the Ipo Curve from FrontWheel for the RearWheel, with only one Ipo Curve for the six Floor-Wheels, and one for the MiddleWheel if its size is not the same as FloorWheels.



Do you remember the "OIRA or AFAPIRA" acronyms? Well, the Ipo Speed for the Path is a straight line and doesn't really count. So the only object to animate with an Ipo Curve is the Empty, Master. In Frame 1 (Shift+?) select Master, Insert a Keyframe for Location (I, 1), go to Frame 51 (?????), move Master along X axis, and Insert a Keyframe for Location (I, 1). In the Ipo window you can select and delete the Ipo LocY and LocZ, and/or you can edit and adjust jsut the Ipo LocX.

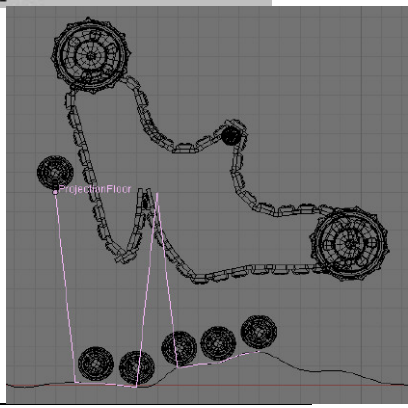


If the Mesh SoftFrame acts strangely, go in the Panel "Collision" and click on "Free Cache"



Pay attention to the value for End in the same panel if you do an animation longer than 250 frames.

If the Mesh ProjectionFloor acts strangely, edit the Mesh Floor, select all, and extrude a tiny little bit (E, 1, Z, -0.01). If no there's result, try to delete the Subsurf Modifier, or try to apply it.

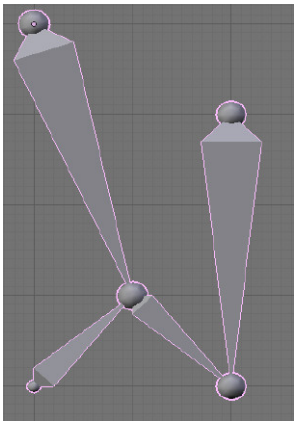


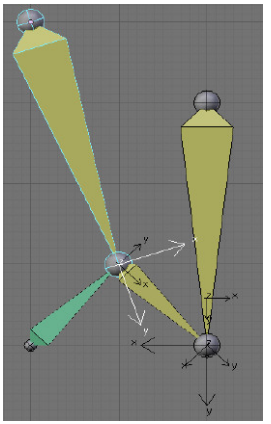
by Alain Mathez

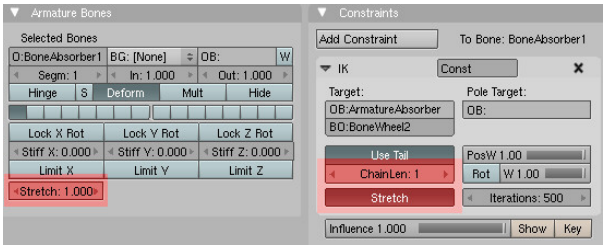
Note for the debug warriors (maybe like me, I don't know) *: If you find why the Shrinkwrap Modifier does this jump, please let me know how to correct it.*

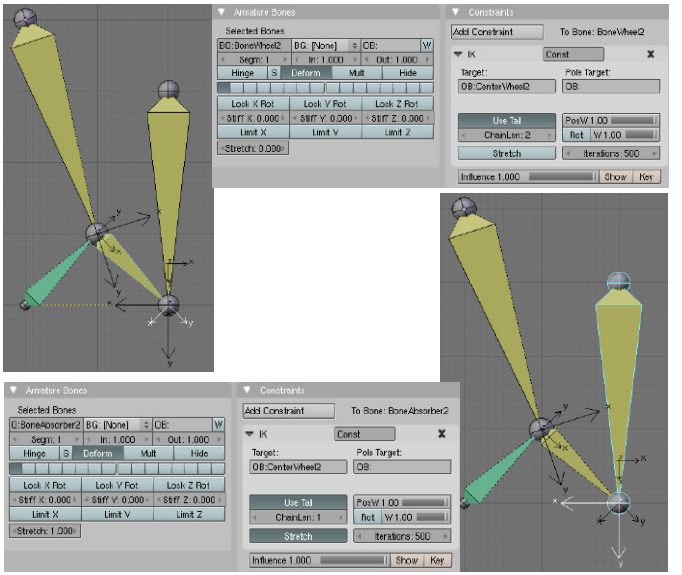
Remember you can set the fine tuning with the Influence of the Copy Rotation constraints from CenterFrontWheel, AngleFrame, CenterRearWheel, and you can choose between Copy Rotation and TrackTo for OrientateFrame.

11.Layer 11-13 : the shock absorber. Let us do a piston, a bellows and a spring. On the Layer 11: an Armature.







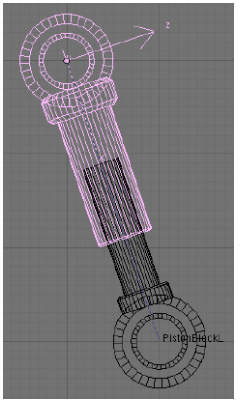


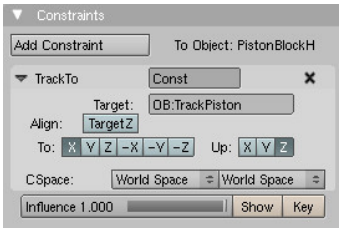
On the Layer 12: some Empties, to use as the target for IK or TrackTo (The choice is yours).

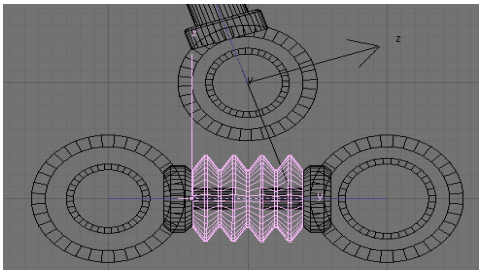
On the Layer 13: the objects. At first the piston, Block High has a TrackTo in the direction of Block Low, and Block Low has a TrackTo in the direction of Block High.

(Editor's note: This introduces a cyclic dependency which Blender has trouble resolving. If possible, it is best to avoid such setups altogether to be safe. However, if such setups are used, be aware that some lag may occur in some parts of the rig, making it unsuitable for renderfarm use.)

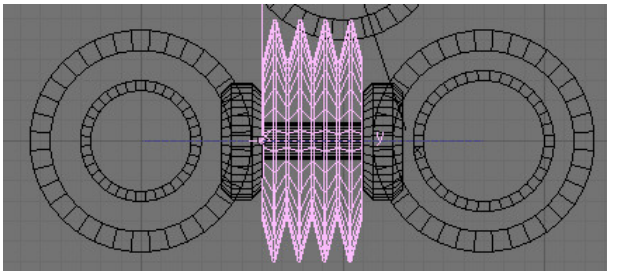
by Alain Mathez

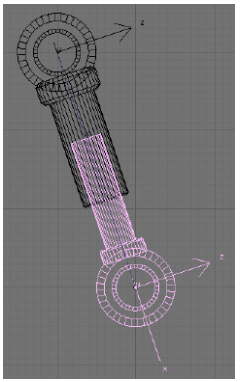


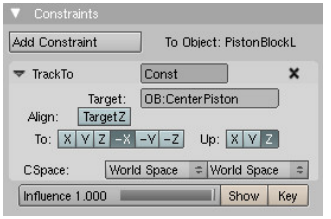


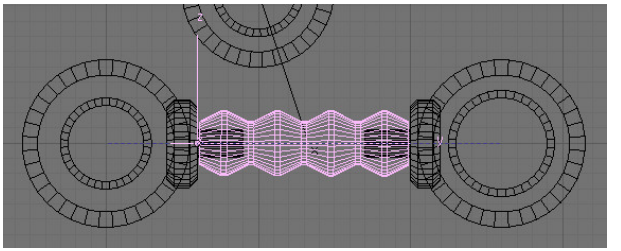


The effect is simple but the illusion is perfect....

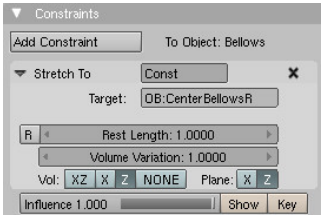






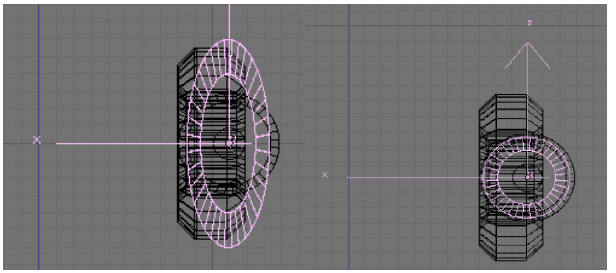


Now the bellows. Block Left and Block Right are now tracking each other like a piston. The bellows will need a Stretch To Constraint.



...as long you don't switch to the Side View. If the bellows are deformed (don't ask me why) just adjust the Z size (S, Z, MouseMove).

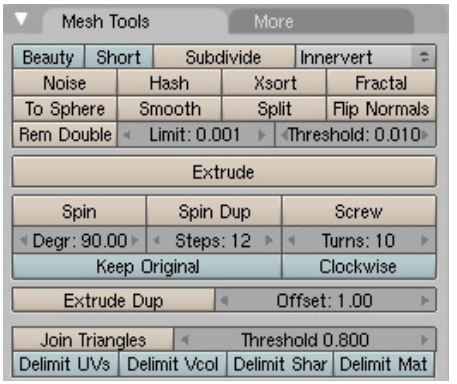
by Alain Mathez



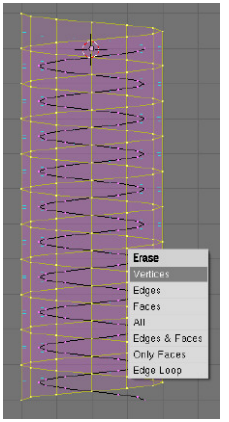
If the orientation of the bellows is wrong, show its axes in the panel “Draw” (in the "Object" buttons). Rotate all the vertices in Edit Mode, with the cursor at the origin of the object and Pivot Mode set to 'Around the Cursor'.

Finally the spring. This one will be vertical. The TrackTo Constraint needs to have an axis pointing upwards, so in this case there is a conflict between the target directions and the up pointer. We will use a Bone, with a IK Constraint to orientate the Up and Low Blocks, and each Block will have a Copy Rotation of the Bone. Almost as easy as for the piston.

How to model a spring? Easy, but we want to animate it! Add a Plane, place the Cursor at its center, in Front View let only an Edge and a Vertex. Select all, and press



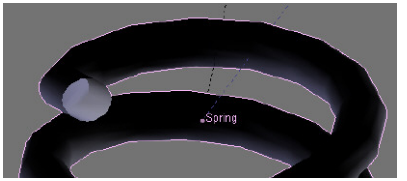
Screw in the panel “Mesh Tools”.



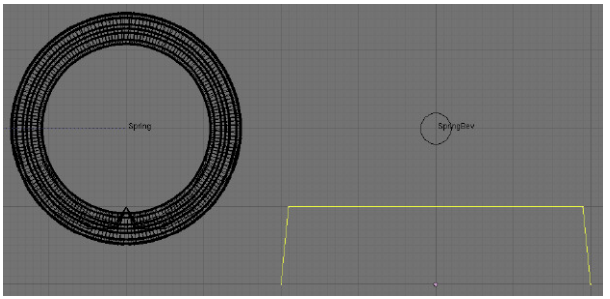
Erase the cylinder and keep the spring.

Select all, go to the Menu Mesh › Scripts › Edges to Curve. The script did a copy from the Mesh, and we have now a new Curve. Edit this Curve, select all the Vertices, in the panel “Curve Tools” click on Nurb, Endpoint U, Order U: 3, Weight 1, Set Weight. Go back to Object Mode. Add a Curve NURBS Circle, name it SpringBev, reduce its size. Select the Curve Spring (not the Mesh, you can delete this one), give SpringBev as Bevel Object in the panel “Curve and Surface”. Nothing happens, so edit the Curve Spring, select all, set the radius to 1 (W, 4, Enter). Special thanks to Zeauro from the BlenderClan for this trick.

But the wire of a spring is not a hollow tube, so we need another NURBS Curve as Taper Object. Note that the size of Spring-Taper is 0.1 in Object Mode.

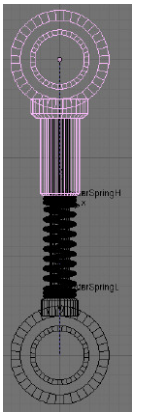


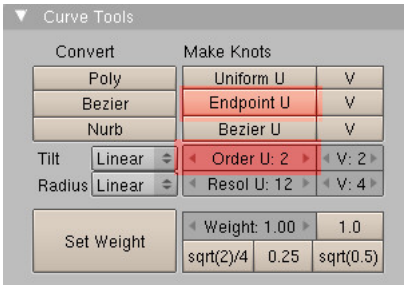
by Alain Mathez



SpringBlockH, Child of FixeAbsorber1 (the main Empty for the absorber)

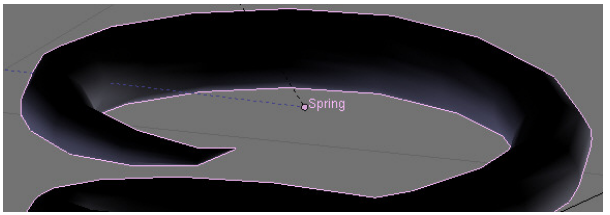
receives a Copy Rotation Constraint targetting BoneAbsorber2 from ArmatureAbsorber. Remember, a TrackTo Constraint is not possible because Object has to track vertical down.

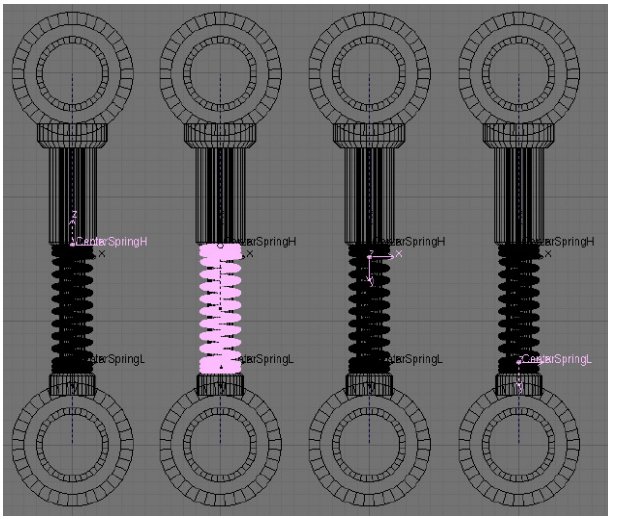




Empty CenterSpringH, child of SpringBlockH.

- CurveSpring, child of CenterSpringH.
- Empty ScaleSpring, child of CenterSpringH.
- CenterSpringL, child of SpringBlockL.





Resize a bit the upper and the lower part of the spring, you know, to obtain something like… a spring.

Note for the tired people (like me, it's late) *: sorry just a bad joke.*

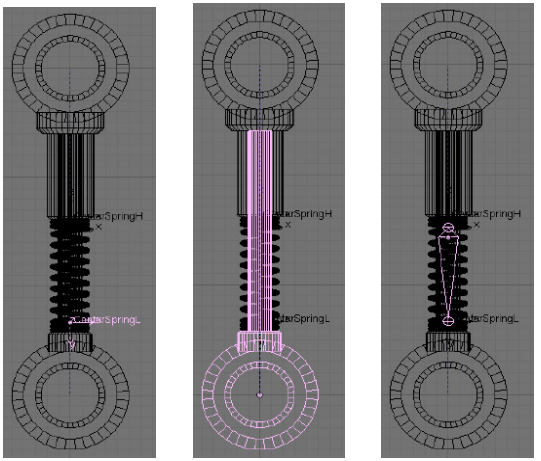
Construction, from top to bottom.

by Alain Mathez

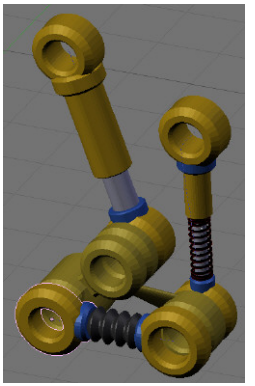
SpringBlockL, child of the Empty CenterWheel2.

BoneSpring from ArmatureSpring, child of CenterSpringH.

BoneSpring has an IK Constraint on CenterSpringL and Stretch enabled. The high (local Y axis) of BoneSpring is precisely equal to the deformable part of the spring. Add a Copy-Scale Constraint (on Y-axis only) to the Empty ScaleSpring. ScaleSpring is oriented like BoneSpring of course, and placed at the begining of the deformable part of the spring. Select all the Vertices from the deformable part of the spring, select ScaleSpring, make a Hook. Select the lower un-deformable part, select CenterSpringL, make a Hook.



So, like you see, rigging a dynamic spring is not too difficult. But once again this method is not mine, I derived it from here and simplified it (I guess) and improved (I hope). So, "Dude, all you have to do is...already done!"



Well, that's it. Add some mecanical elements to complete the absorber. Of course a tank have no such absorbers visible and fragile, but 3D is not the real world, so... why not ?

12.The challenge

Now if you want to make a vehicle moving on a bumpy ground, you will face bigger problems. The second caterpillar track can't ride for itself. Both must have the same tilt and keep the distance between them constant, because this distance is the vehicle.

Well, who can do it better, is more simple and faster? It is possible, I'm sure. But I didn't already found the solutions. And don't forget the "OIRA or AFAPIRA" principle.

If you're better than me (I'm sure it's possible, because I succeed on doing this almost every day ;-) and find a better way, or if you have questions, remarks, ideas, suggestions,... feel free to contact me at alain@bktr.ch

My favorite language is French, but I can decode English (please simple words only ;) und auch ein bisschen Deutsch (da noch mal bitte nicht zu komplizierte Wörter verwenden).

Have fun!

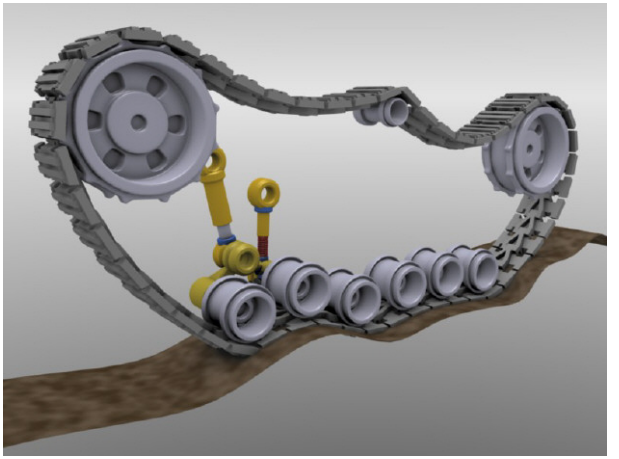
by Alain Mathez

13.Example files

In the example file "CaterpillarSoftTrack.blend" you will find the object described in chapter 1 to 11, a tank downloaded from http://dmi.chez-alice.fr and a lot of Cameras fixed to each important part of the caterpillar. Some objects have still a French name, sorry.

In the example file "TrackLinkApproximation.blend" there are the 2 methods described just before chapter 8 to make links follow a Path. Other ideas to do this?

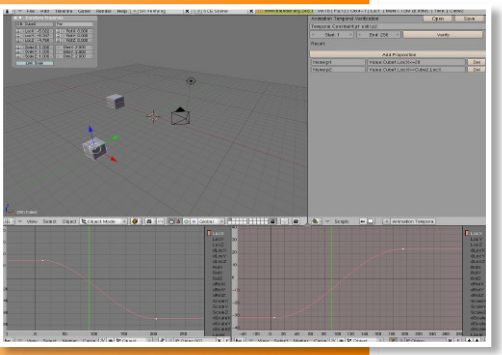
In the example file "LimitePathPyDriver.blend" there are a lot of test with Ipo Speed and PyDriver you can play with.



14.Gratitude

Special thanks to the BlenderClan, Meltingman (my spiritual Master), Traven (the man who knows everything I ignore) Zeauro (I have questions, he has answers), and to my honey Coralie who slept and watched the television alone for two weeks while I wrote this. And THANK YOU to every people who makes Blender growing.

by Alain Mathez



By Hamed Zaghaghi

Introduction

Temporal verifiers are mathematical tools that verify a temporal model against a temporal property, (i.e. properties that talk about time), and answer the question, "Does this temporal model satisfy these temporal properties?".

This concept is used by many researchers in the field of Software and Hardware Engineering, and I bring it to the world of computer animations.

"*Animation Temporal Verification*" is a new and novel approach in computer graphics that lets us:

- verify animations against temporal properties that can be extracted from story check temporal constraints for animations.
- A temporal property is a temporal logic formula that will described later in this text. For example:

Always (Cube1.LocX>=100.0)

That means X part of Cube1 location should always be greater than or equal to 100.0.

Definitions

Here you can see some fundamental definitions for understanding what we can do with this tool to improve the time of checks and verifications.

Linear Temporal Logic

Linear temporal logic (LTL) is a modal temporal logic with modalities referring to time.

Syntax

LTL is built up from a set of propositional variables p1,p2,..., the usual logic connectives *(not, and, or, implies)* and the following temporal modal operators *Next, Globally or Always, Eventually or Future, Until.*

Semantic

An LTL formula can be evaluated over an infinite or finite sequence of truth evaluations and a position on that sequence. An LTL formula is satisfied by a sequence if and only if it is satisfied for position 0 on that sequence. The semantics for the modal operators is given as follows:

- Next φ holds at position i, if exists position i+1 and φ holds at position i+1
- ψ Until φ holds at position i if φ holds at some position j, and ? has to hold at all positions in range [i to j).
- Eventually φ or Future φ, holds at position i, if there was a position j>i and φ holds there. In other words Eventually φ holds if and only if True Until φ holds at position i.
- Globally φ or Always φ, holds at position i if and only if Not Eventually Not φ holds at that position, in other words Globally f holds at position i if φ holds at all positions.

by Hamed Zaghaghi

Important Properties:

We can construct various temporal properties with linear temporal logic, but there are two main types of properties that can be expressed using this logic:

- *Safety properties (Globally Not φ): usually state that something bad never happens.*
- *Liveness properties (Globally Eventually φ) or Globally(ψ Implies Eventually φ): state that something good keeps happening.*

See [1,2] for more information about temporal logic.

Implementation

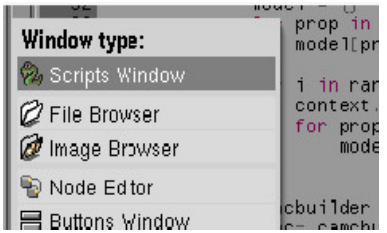
This feature is implemented completely using Python and is available for Blender using the scripts menu after installation.

You can get the Python script and it's required module, pycamc, from http://launchpad.net/camc.

You can find more information about it's licence and how you can install it from above project page.

How To Use It

After installing it, open blender and open the script window in some area in blender. See Fig.1.

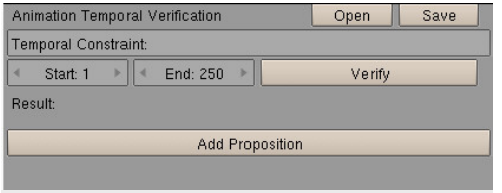


Opening script window

Then in the script window select the Animation Temporal Verification item from the Animation sub menu of the Script menu. See Fig. 2.

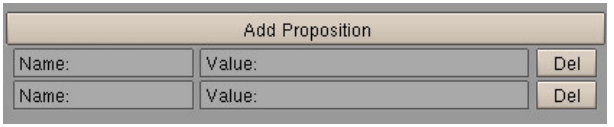
Opening Animation Temporal Verification

in Fig. 3. and works as follows:



Empty Form

- Open/Save Buttons opens/saves information from/to a file.
- Temporal Constraint text box, should contain a temporal property
- Start and End number boxes, shows start frame and frame of verification.
- Verify button, starts verifying the Temporal Constraint.
- Add Proposition button adds new text boxes to the form so you can input a relational expression as a proposition there. See Fig. 4

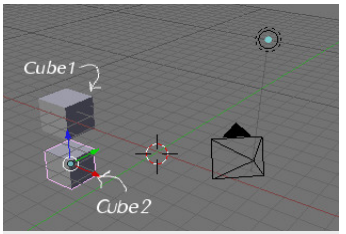


Add Proposition

- Name text box specifies the name of the proposition and can be used in a Temporal Constraint
- Value text box specifies the boolean expression related to the proposition name
- Del button, deletes the proposition in it's row

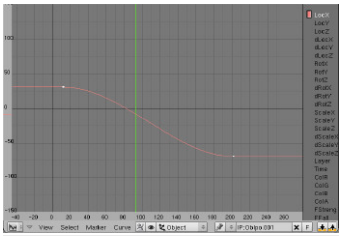
Example

Here we want to verify a simple temporal behaviour in an animation. The sample scene contains two Cubes, named Cube1 and Cube2, a Lamp and a Camera, see Fig. 5.



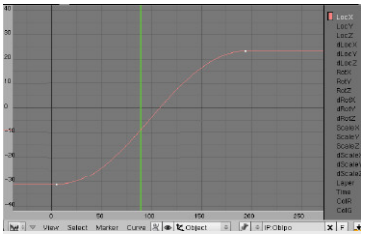
Sample Scene

Now consider that we set up the scene so Cube1 moves in the X direction from lower values to higher values of X and Cube2 moves in the X direction too but from higher values to lower values of X. See Fig. 6 and Fig. 7.



Cube2 IPO for *LocX*

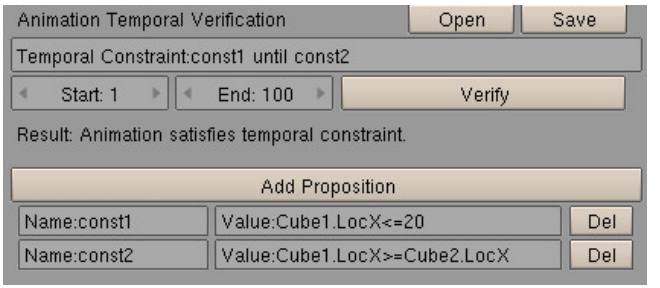
Furthermore, we want that the X part of the Cube1 location is smaller than or equal to 20 until the Cube2 X location reaches the Cube1 X location.



Cube1 IPO for *LocX*

After setting up the scene, creating IPOs etc. we want to check that the above statements are true according to this scene. To do so:

1 Add two propositions
   - proposition 'const1' with value 'Cube1.LocX<=20'
   - proposition 'const2' with value 'Cube1.LocX<=Cube2.LocX'
2 Add the following constraint in Temporal Constraint text box
   - 'Const1 Until Const2'
3 Press Verify button to start verification process.



Example Verification Form

by Hamed Zaghaghi

Final Words!

First I should thank my co-worker, Babak Salimi, who helps me in the theoretical side of work, and then thanks to you for reading this article.

Please get it, test it, and report its bugs!

To get

download it from https://launchpad.net/camc

or get latest trunk source with

bzr branch lp:camc

To report

report bugs here

https://bugs.launchpad.net/camc/1.0/+bugs

email me

hamed.zaghaghi@gmail.com or zaghaghi@ut.ac.ir

References

1.Amir Pnueli. The temporal logic of programs. In FOCS, IEEE, 1977.

2.http://en.wikipedia.org/wiki/Linear_temporal_logic

by Hamed Zaghaghi

by Sandra Gilbert

Armatures provide Blender with a very powerful character animation tool, but armatures can help you animate far more than just characters. I know I have personally used armatures to help animate all kinds of objects. But wouldn't you know it, when I sat down to write this article, I couldn't think of even one good example (my memory seems to be slipping with age).

Still not a problem; that's what the search function is for. I typed in "basic rigs" and was immediately given numerous examples of cool and fun ways to use armatures. This is by no means an all-inclusive list, but it should give you enough ideas that you can come up with even more ways to use armatures in your own project.

I found several great examples in BSoD: Introduction to Rigging.

Just a quick side note before I reveal the genius ideas. Robert Christian (wavez) did an excellent job explaining all about rigging. So if you still haven't read it, I seriously encourage you to do so.

Okay, first up, you can use armatures to produce a great "Squash n' Stretch" effect. This is an ingenious little rig. It is composed of just two bones and four constraints. Here's the ingenious part. One of the constraints, is a "Stretch To" constraint, which will automate the squashing and stretching for you. In my opinion, it's very cool. (Hmm... now I want to squash something!)

Next Robert shows you how to create a dancing palm tree. Now I'm sure you don't really have a huge need for dancing trees (or maybe you do), but think about it, this could be adapted for so many things.

As I wandered in my research, I found a great tutorial for "Making a Spring Rig"

in Blender, over at the Blender Underground site. It takes a little bit to set up, but once you are done, you have a nice dynamic moving spring.

JiriH created a great little "Squash and stretch ball rig", to give beginning riggers something to study and play with.

And of course armatures are excellent for creating Mechanical Rigging. This tutorial shows you how to rig a robotic creature, including how to set up pistons and hydraulics.

And who can resist having a Dancing Beer rig? The guys at ProMotion Studios were obviously having a little too much fun with their current animation project Kajimba, but I don't think any of us will be complaining, since we get to play with it.

Well that about wraps up my little search survey of fun things to do with armatures. I hope it gave you some interesting ideas of your own to try out. ■

Hoo and Ray for the Mesh Deform Modifier!

by Sandra Gilbert

I have to be honest, rigging is not my favorite task. All those bones and constraints that have to be set up just so, or they don't work. Plus the endless testing and tweaking is enough to make me pull my hair out. The only thing I enjoy less than rigging is actually attaching the rig to my object (or character).

So I was beyond tickled when the coders gave us Bone Heat. The ease of use made a horrible task, relatively painless. But as good as it is, for anything but a fairly simple character, you still have to do some tweaking to get your character to deform right. I don't enjoy tweaking, so guess how happy I was when our amazing coders gave us the Mesh Deform Modifier... Very!

Okay, so just what is the Mesh Deform Modifier and what do you use it for? In some ways it works like a Lattice, in that you use it to change the shape of a mesh (Object, character etc.). The Mesh Deform Modifier allows any closed mesh (of any shape, not just the cube/grid shape of a Lattice Modifier) to act as the deforming cage around another mesh.

To use Mesh Deform:

- Create a low resolution cage to fit around your character (or object)
- It must form a closed cage around the character (or object) to be deformed
- Add a Mesh Deform modifier to the character mesh
- In the Ob: field, enter the name of your low resolution cage
- Push the bind button

Couldn't be easier. It will take anywhere from a few seconds to a few minutes to bind your cage to your mesh.

Now that your mesh deform modifier is bound to the character, you can apply an armature deformer (modifier) to the cage and then use Bone Heat or Vertex groups, just as if if were a higher resolution mesh.

Remember: the armature deforms the cage, the cage controls and deforms the higher resolution character mesh.

Tips for better results:

- Use triangles (not quads) in the Cage mesh
- If you prefer to work with quads, just press Alt + T in Edit mode to convert to triangles once the cage is built and prior to binding.
- Start with low levels of precision and increase as needed if distortion appears
- High precision levels can cause the binding process to take a very long time or even cause Blender to crash.
- Distortion can be minimized by moving the cage further out from the Character mesh (unbind before trying to move the cage from character).
- If any part of the character gets left behind when animating, there are vertices and or faces outside of the cage, unbind, adjust the cage, re-bind.
- Any/all changes to the cage itself will require that you unbind the cage from the character first, or the character will be affected/changed too.

by Sandra Gilbert

The Mesh Deform Modifier and Armature combo can be adjusted and tweaked just like the Character Mesh and Armature combo you are used to. Bone heat (as mentioned) works well with the Mesh Deform Modifier and because the cage is generally of a much lower resolution, it requires less fiddling with and tweaking than a high resolution mesh.

Yay! That means less work for me!

As an additional plus, the Mesh Deform Modifier can even be used as a modelling tool. Any edits/changes to the cage (in Edit Mode) while bound to a mesh object will cause changes and deformations to the mesh object as well as the deforming cage. Then you just click the apply button to make the changes permanent.

There are a few things about the Mesh Deform Modifier that are not overly obvious. Listed below are some of those issues and features (taken from the Blender User Manual).

Mode of Operation

Alterations made to the Deform Mesh Cage will only be reflected in the Deformed Object when the cage is in Edit Mode, when in Object mode the cage can be scaled and distorted but it will not effect the Deformed Object.

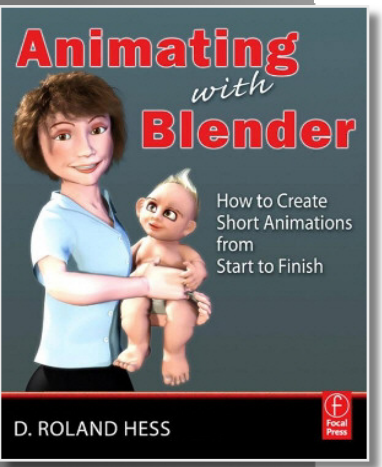
Deform Mesh Cage Location AFTER Binding

While a Deform Mesh Cage is being bound to a Deformed Object the cage must surround all the parts of the Deformed Object you wish to be affected by the cage. Once the Deform Mesh Cage has been bound it can be moved away from the Deformed object in Object Mode. When you then switch the Deform Mesh Cage back to Edit Mode and alter its shape, it will alter the Deformed Object even when it is not directly surrounding it.

Distance from Deform Mesh Cage & Deformed Object

Distance between the Deform Mesh Cage and the object to be deformed (Deformed Object) has an influence on the amount of change imparted to the Deformed Object when the Deform Mesh Cage is altered (when in Edit Mode). When the Deform Mesh Cage is further away from Deformed Object, then the amount of change imparted to the Deformed Object is less and less local to a specific area of the Deformed Object. When the Deform Mesh Cage is closer to the Deformed Object the amount of influence upon the Deformed Object is greater and more local to a specific area on the Deformed Object.

BOOK REVIEW: Animating with Blender (How to Create a Short Animation From Start to Finish)



★★★★★

5/5

So you want to create the next "smash" animated short. You have an idea. You have Blender. You are all set.

Or are you? Most ambitious (and non ambitious) animation projects fail due to lack of planning and/or lack of what it really takes to actually start, and even more important, finish a short animation.

Animating with Blender was written to answer all of the questions you have about creating an animation. Better yet, it answers questions you haven't yet considered.

Roland Hess, author of Animating With Blender, takes you step-by-step through the animation process.

After an informative discussion on why following the work-flow is important, Roland starts walking you through the steps to create your own animated short. Using his own short animation "The Beast" as a learning tool, he shows you the importance of a good story and how to set up both your storyboards and story reel.

As you work your way through the succeeding chapters, you not only learn about the next stage in the process, but the best tools to get it done. Each chapter builds on the previous just as each actual stage of production builds on what went before.

Along the way, you learn about character and set design, blocking out shots, and matching up your sound track. As you progress through the book, Roland shares valuable tips and tricks to make the whole thing go smoother.

Roland has a great writing style that I have always enjoyed. I found the book to be casual and friendly as well as extremely entertaining. In my opinion, considering the massive amount of information contained in its 352 pages, the humorous touch makes this an enjoyable as well as informative book to read.

Since I have yet to create any animation worth mentioning, the whole book proved to be a huge learning experience. Below I have listed some of the areas that really caught my attention:

- The tips on story importance, how to refine and polish your story before you start
- His organization suggestions for preventing future headaches.
- The importance of creating your storyboards with the correct aspect ratio, so that your shots line up the way you planned.
- Great tips for using libraries and linking them properly
- Use of the Mesh Deform Modifier, as well as great rigging tips
- The eye rigging set up is the easiest to set up that I have ever seen
- Great Action Editor tips and explanations of features I never even knew about
- Rendering tips for the shortest render times while still getting good quality
- Getting it all put together
- All the wonderful workarounds

My overall reaction: this is a beautifully written guide to creating an animation. It is filled with useful tips and tricks that can be used for not only animation projects, but other projects as well. It is well written in reader friendly language and uses humor to illustrate and explain difficult concepts.

The book has full-color screen shots and renders, so you can clearly see what is being discussed. This is a valuable addition to the Blender knowledge base, and will be an often used reference in my own projects. ■

Book Review by Sandra Gilbert | **Paperback:** 368 pages | **Publisher: Focal Press:** Pap/Dvdr edition (September 30, 2008) | **Language:** English
ISBN-10: 0240810791 | **ISBN-13:** 978-0240810799 | **Product Dimensions:** 9.2 x 7.5 x 0.8 inches

JetPack Girl - Zooly

Magic Painter - Zooly

Botella Llena - Raul Silva

Want to write for BlenderArt Magazine?

Here is how!

1. We accept the following:

- Tutorials explaining new Blender features, 3dconcepts, techniques or articles based on current theme of the magazine.
- Reports on useful Blender events throughout the world.
- Cartoons related to blender world.

2. Send submissions to sandra@blenderart.org. Send us a notification on what you want to write and we can follow up from there. (Some guidelines you must follow)

- Images are preferred in PNG but good quality JPG can also do. Images should be separate from the text document.
- Make sure that screenshots are clear and readable and the renders should be at least 800px, but not more than 1600px at maximum.
- Sequential naming of images like, image 001.png... etc.
- Text should be in either ODT, DOC, TXT or HTML.
- Archive them using 7zip or RAR or less preferably zip.

3. Please include the following in your email:

- Name: This can be your full name or blenderartist avtar.
- Photograph: As PNG and maximum width of 256Px. (Only if submitting the article for the first time)
- About yourself: Max 25 words .
- Website: (optional)

Note: All the approved submissions can be placed in the final issue or subsequent issue if deemed fit. All submissions will be cropped/modified if necessary. For more details see the blenderart website.

Upcoming Issue 'Theme'

Issue 19

Make it, Bake it, Fake it

- Learn about various tools in Blender that require you to baking the result.
- Render baking options.
- Water
- Physics
- Cloth, Soft bodies and more...

Disclaimer